(2)

A Microcomputer Pollution Model for Civilian Airports and Air Force Bases MODEL DESCRIPTION

U.S Department of Transportation
Federal Aviation Administration
Office of Environment and Energy
Washington, D.C. 20591

United States Air Force
Engineering Services Center
Tyndall Air Force Base, Florida 32403

August 1988

H.M. Segal
P.L. Hamilton

FAA FFA-EE-88-4
USAF ESL-TR-88-53

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
(FAA) FAA-EE-88-4 (USAF) ESL-TR-53		

4. Title and Subtitle	
A MICROCOMPUTER POLLUTION MODEL FOR CIVILIAN AIRPORTS AND AIR FORCE BASES -- MODEL DESCRIPTION	5. Report Date AUGUST 1988
	6. Performing Organization Code
7. Author(s) H. M. SEGAL P. L. HAMILTON *	8. Performing Organization Report No. DOT/FAA
9. Performing Organization Name and Address FEDERAL AVIATION ADMINISTRATION OFFICE OF ENVIRONMENT AND ENERGY 800 INDEPENDENCE AVENUE, SW. WASHINGTON, DC 20591	10. Work Unit No. (TRAIS)
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered MODEL DESCRIPTION
	14. Sponsoring Agency Code AEE-30(FAA);AFESC/RDV.S(USAF)

15. Supplementary Notes

* MSI SERVICES INCORPORATED
1925 "K" STREET, NW.
WASHINGTON, DC 20006

16. Abstract

This is one of three reports describing the Emissions and Dispersion Modeling System (EDMS). All reports have the same main title—A MICROCOMPUTER MODEL FOR CIVILIAN AIRPORTS AND AIR FORCE BASES--but different subtitles. These subtitles are:

(1) USER'S GUIDE - ISSUE 2 -----------(FAA-EE-88-3/USAF-TR-88-54)
(2) MODEL DESCRIPTION ---------------(FAA-EE-88-4/USAF-TR-88-53)
(3) MODEL APPLICATION AND BACKGROUND -(FAA-EE-88-5/USAF-TR-88-55)

EDMS is a complex source emissions/dispersion model for use at civilian airports and Air Force bases. It operates in both a refined and a screening mode and is programmed for an IBM-XT (or compatible) computer. This report--MODEL DESCRIPTION--provides the technical description of the model. It first identifies the key design features of both the emissions (EMISSMOD) and dispersion (GIMM) portions of EDMS. It then describes the type of meteorological information the dispersion model can accept and identifies the manner in which it preprocesses National Climatic Center (NCC) data prior to a refined model run. The report presents the results of running EDMS on a number of different microcomputers and compares EDMS results with those of comparable models. The appendices elaborate on the information noted above and list the source code.

17. Key Words	18. Distribution Statement
POLLUTION, AIR POLLUTION, DISPERSION MODEL, EMISSIONS MODEL, DATA BASE, MICROCOMPUTER	THIS DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED		

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

1 - INTRODUCTION

1.1 BACKGROUND

Almost any expansion of operations or facilities at airports or airbases requires some type of an environmental assessment. In the area of air quality, this assessment usually involves the use of two models--one to prepare an inventory of emissions and the other to calculate the concentrations caused by these emissions as they disperse downwind.

Models to perform these tasks were developed in the early 1970's by both the United States Air Force (USAF) and the Federal Aviation Administration (FAA). The USAF developed the Air Quality Assessment Model (AQAM) (Rote, et al., 1975), and the FAA developed the Airport Vicinity Air Pollution model (AVAP) (Wang, et al., 1973). However, these models are becoming obsolete--they are expensive to operate, tedious to enter data into, and require a fully-qualified scientist or engineer to use.

The introduction of modern microcomputers into the workplace has made it possible to simplify the modeling task considerably. However, a prime question to be asked before developing a new model is whether to develop a separate modeling system for each agency, as was done with the earlier AVAP and AQAM systems, or whether to combine FAA and USAF requirements into a single system.

Consultations between the FAA and the USAF indicated that it would be both feasible and cost effective to develop a single modeling system that both agencies could use. The system that resulted from these consultations is called the Emissions and Dispersion Modeling System (EDMS). It overcomes the limitations of the earlier models by incorporating the new "Simplex A" (Segal, 1981) and the Graphical Input Microcomputer Model (GIMM) (Segal, 1983) technology. The EDMS also employs a modern commercial data base (Condor, 1983) to enter, store, and preprocess information. These features permit a lay person to perform a modeling task that had previously been reserved for a scientist or engineer.

1.2 APPROACH

The development of EDMS started in the late 1970's when the feasibility of using a microcomputer to perform a mainframe dispersion modeling task was first established. In 1979, an algorithm to trace the dispersion of pollutants from aircraft that were accelerating during takeoff was successfully incorporated into a HP-97 programmable desk calculator. The resulting model was called Simplex "A".

With the introduction of personal computers in the early 1980's, the HP-97 code was reprogrammed for an Apple II+ microcomputer. This approach led to the original GIMM--Graphical Input Microcomputer Model. Because of its more powerful Apple computer, GIMM could act as a complex source model to process the dispersion of emissions from roadways, parking lots, and powerplants as well as from accelerating aircraft.

Realizing the effectiveness of GIMM in meeting both FAA and USAF needs, the FAA and the USAF issued a memorandum of understanding (MOU) to formally blend the efforts of both agencies. This MOU documented the need for a single FAA/USAF microcomputer model to evaluate air quality at both airports and airbases. This model--the Emissions and Dispersion Modeling System (EDMS)--incorporates the emissions and dispersion algorithms of the original GIMM that have been speeded up and processed through a commercial data base. EDMS was completed in 1985, and its code and User's Guide were released to the general public in December 1985 as report FAA-EE-85-4/ESL-TR-85-41.

Since that time, major modifications have been made to the original EDMS to ennance its usability and incorporate an integral dispersion model into its code. A prototype of this expanded model was completed in 1986 and was used to analyze air quality at Stapleton International Airport in conjunction with the building of a new runway for that airport.

Since 1986, the prototype EDMS has been incorporated into the main EDMS system, and the final model has been submitted to the EPA as an agenda item for the Fourth Conference on Air Quality Modeling. The technical features of this model are described in this document.

2 - MODEL DESCRIPTION

2.1 GENERAL

EDMS is a dispersion model with an emissions front end. It can process line, point, and area sources at an airport or airbase and can operate in a refined or screening mode.

The emissions portion of EDMS receives emissions information entered through its Condor data base (Condor, 1983) and converts this information into emission rates from which an emissions inventory can be prepared.

The dispersion portion of EDMS adds meteorological inputs to this emissions information and produces a report of concentrations at specified airport locations. The model can operate in a refined or a screening mode. In its refined mode, it automatically processes National Climatic Center (NCC) weather files. In the screening mode, it processes weather data that has been manually entered on an hour-by-hour basis.

Figure 1 shows the elements of the emissions model, and Figure 2 shows the manner in which the emissions and dispersion models interface.

The emissions model, EMISSMOD, incorporates a modern data base, (Condor, 1983). The dispersion model, GIMM, incorporates the dispersion algorithms of the original GIMM which have been speeded up to enable refined model operation.

2.2 EMISSIONS MODEL (EMISSMOD)

EMISSMOD accepts emission and coordinate information for the following airport sources:

I Miscellaneous point sources (Facility sources)

- Powerplants
- Heating plants
- Fuel storage tanks
- Training fires

II Roadway sources

III Parking lot sources

IV Aircraft sources

- Aircraft queues
- Aircraft takeoffs

The interrelationship of these sources, with regards to data entry is shown in Figure 3.

Data Flow — Emissions Model

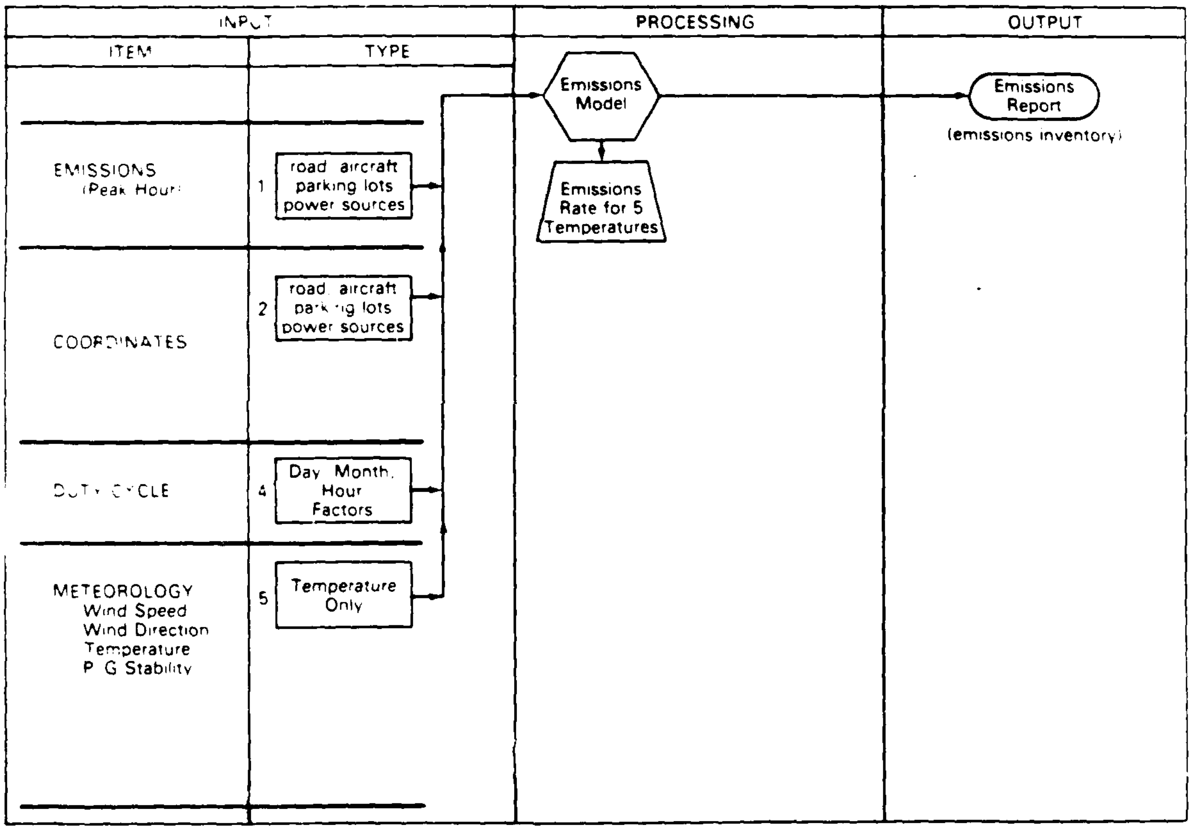
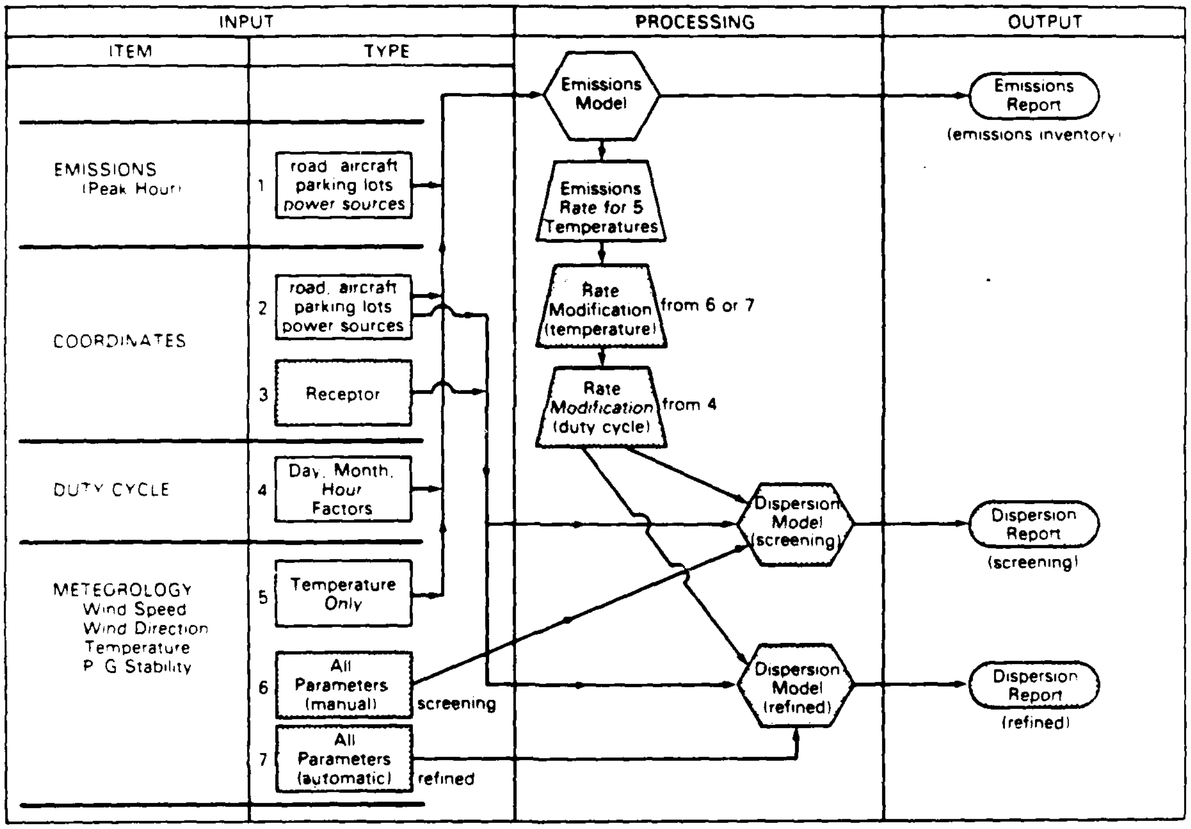


Figure 1

Data Flow — Emissions and Dispersion Model



NOTE Shading represents dispersion model elements.

Figure 2

Organizational Structure

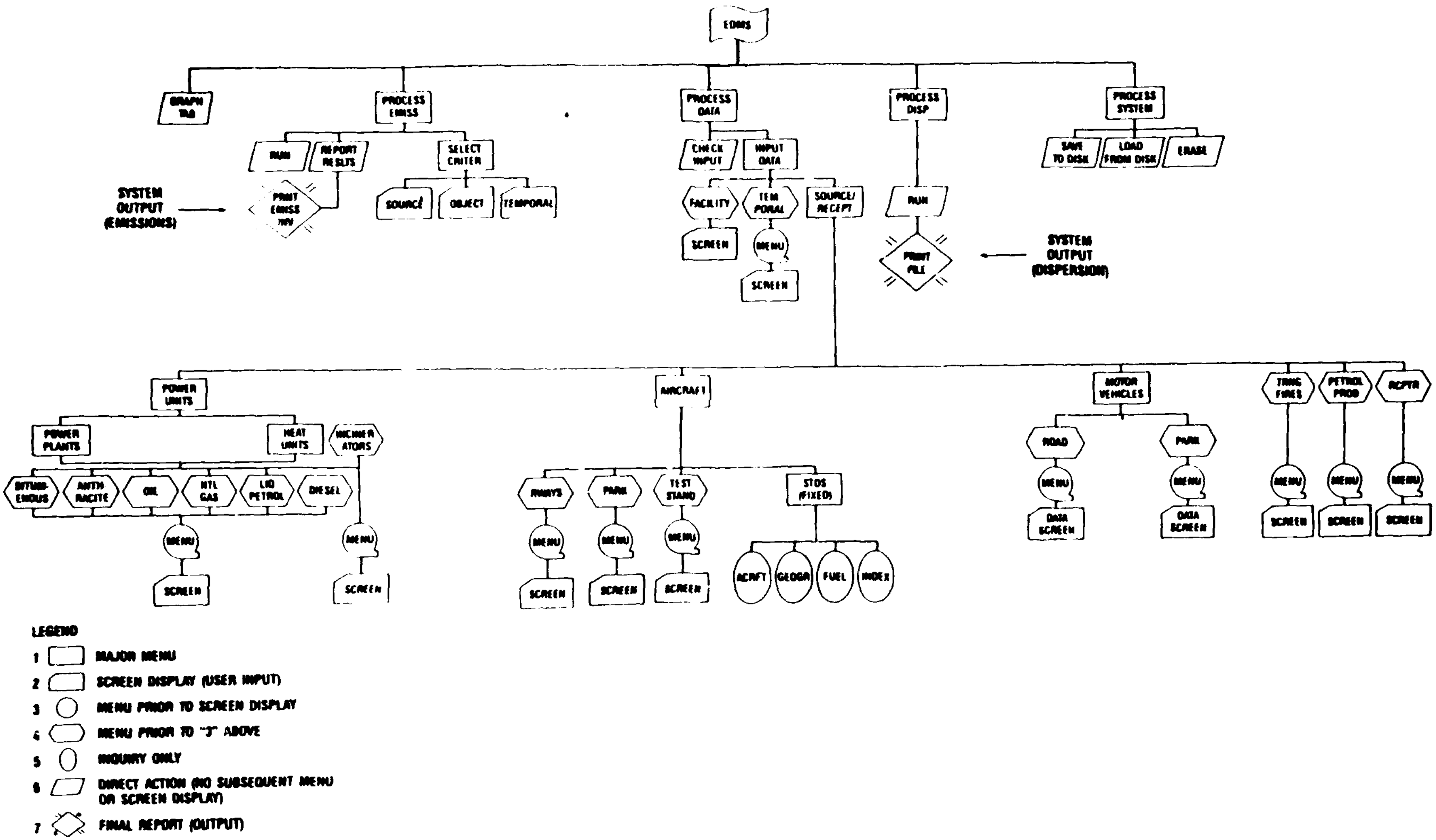


Figure 3

The outputs from EMISSMOD are:

1. An emission inventory report, and

2. Five peak hour emission rates which are used to process motor vehicle emissions information prior to a dispersion model run. These rates, which are calculated for 0, 25, 50, 75, and 100 degrees F., comprise the table lookup matrix for subsequent temperature interpolation of emissions data.

2.3 DISPERSION MODEL (GIMM)

2.3.1 Model Features and Limitations

The EDMS uses the original GIMM code modified to eliminate unnecessary, redundant, and time-consuming calculations. Sinusoidal, exponential, and iterative calculations are replaced with table lookups. GIMM incorporates the following features:

1. Coordinate rotation so that wind is always from the north. This rotation enables one coordinate system to be used for the entire model.

2. A fixed point spacing algorithm. This fixed point algorithm replaces the earlier GIMM algorithm which iterated point spacing to smaller and smaller values.

   This procedure significantly reduced the time required for dispersion calculations.

3. Table lookups to replace exponential calculations.

4. The calculation of an emission rate for that hour of the year when source activity would be at a maximum. Emission rates (and concentrations) for the remaining hours of the year are determined by multiplying this peak hour rate by hourly, weekly, and monthly source activity factors.

The geometry of all sources is broken down into a series of points with one exception, parking lots. These sources are treated as a series of finite lines.

Source dispersion was considered to be Gaussian in nature and the Pasquill-Gifford (P-G) curves from (Turner, 1970), as well as the point and finite lines equations from the same reference, were used in the model design. Initial dispersion conditions reflecting source turbulence can be entered into all models except the roadway model. For this model the initial conditions, as defined by (Rao, et al., 1980)), are incorporated into the program.

To speed up calculation time, algorithms that were shown to have a negligible effect on concentrations or were found to be unrelated to airport pollution were omitted. An inversion height lidding algorithm was not added because source/receptor distances are usually quite small at an airport and the plume usually does not have a chance to reach the base of the inversion. For aircraft, only the takeoff and queue modes of operation were modeled because concentrations from all the other operational modes have proven to be negligible (Yamartino, et. al., 1980

The point and line source equations from (Turner, 1970) were used in all dispersion calculations.

Point source equation:

$$\chi^{(x,y,0;H)} = \frac{Q}{\pi \sigma_y \sigma_z u} \, EXP\left[-.5\left(\frac{Y^2}{\sigma_y}\right)^2\right] EXP\left[-.5\left(\frac{H}{\sigma_z}\right)^2\right]$$

used in the roadway , accelerating aircraft, queueing aircraft, and facility source algorithms

Line source equation:

$$\chi^{(x,0,0;H)} = \frac{2q}{\sqrt{2\pi}\,\sigma_z u} \, EXP\left[-.5\left(\frac{H}{\sigma_z}\right)^2\right] \int_{p_1}^{p_2} \frac{1}{\sqrt{2\pi}} \, EXP\left(-.5\,p^2\right) dp$$

used in the parking lot algorithms

where:

X =	concentration at the receptor	(ug/m3)
x =	source/receptor distance (along plume centerline)	(m)
y =	perpendicular distance between the receptor and the plume centerline in the crosswind direction	(m)
sigma y =	standard deviation of concentration distribution in the crossmind direction	(m)
sigma z =	standard deviation of concentration distribution in the vertical direction	(m)
H =	effective plume height	(m)
q =	source strength per unit distance	(g/m/s)
p =	y/ sigma y	
p1=	y1/sigma y	
p2=	y2/sigma y	

Detailed descriptions of the four submodels are included in Appendix A.

3 - METEOROLOGICAL DATA

3.1 GENERAL

Meteorological conditions may be specified either:

- o Manually, by entering the temperature, P-G stability class, windspeed, and wind direction when prompted interactively, or

- o Automatically, by specifying the range of Julian hours to be processed. The dispersion model will then process the already loaded National Climatic Center (NCC) annual weather data and calculate the concentrations for the hours specified.

When meteorological data are entered manually, the model operates in the screening mode. When these data are entered automatically it operates in the refined mode. These two terms, which are used many times in this report, are explained as follows:

A screening model accepts estimates of the highest expected value for source activity and the poorest expected value for meteorological conditions (from an air quality standpoint). With these inputs the model calculates concentrations. If concentrations exceed the National Ambient Air Quality Standard (NAAQS) for the pollutant in question, a refined analysis is usually indicated.

A refined model simulates real world conditions by processing hourly meteorological and source activity information in a brute force, unbiased fashion.

3.2 METEOROLOGICAL PREPROCESSING

NCC weather data are loaded into the computer before modeling starts. After loading the NCC weather diskettes, a special preprocessor extracts the appropriate parameters from the NCC data, scans these parameters, corrects for missing or questionable data fields, and enters these data into the files required to run GIMM.

The meteorological preprocessor accepts up to one year of NCC hourly surface observations on an IBM PC compatible diskette in card deck 144 format and performs the following operations on them:

- o Determines the P-G stability class.

- o Eliminates those parameters which are not needed in air quality calculations.

- o Converts the NCC data to an integer format. This conversion reduces disk storage demands.

- o Saves the compressed NCC data on a special file for subsequent use by GIMM.

4 - MODEL SPEED

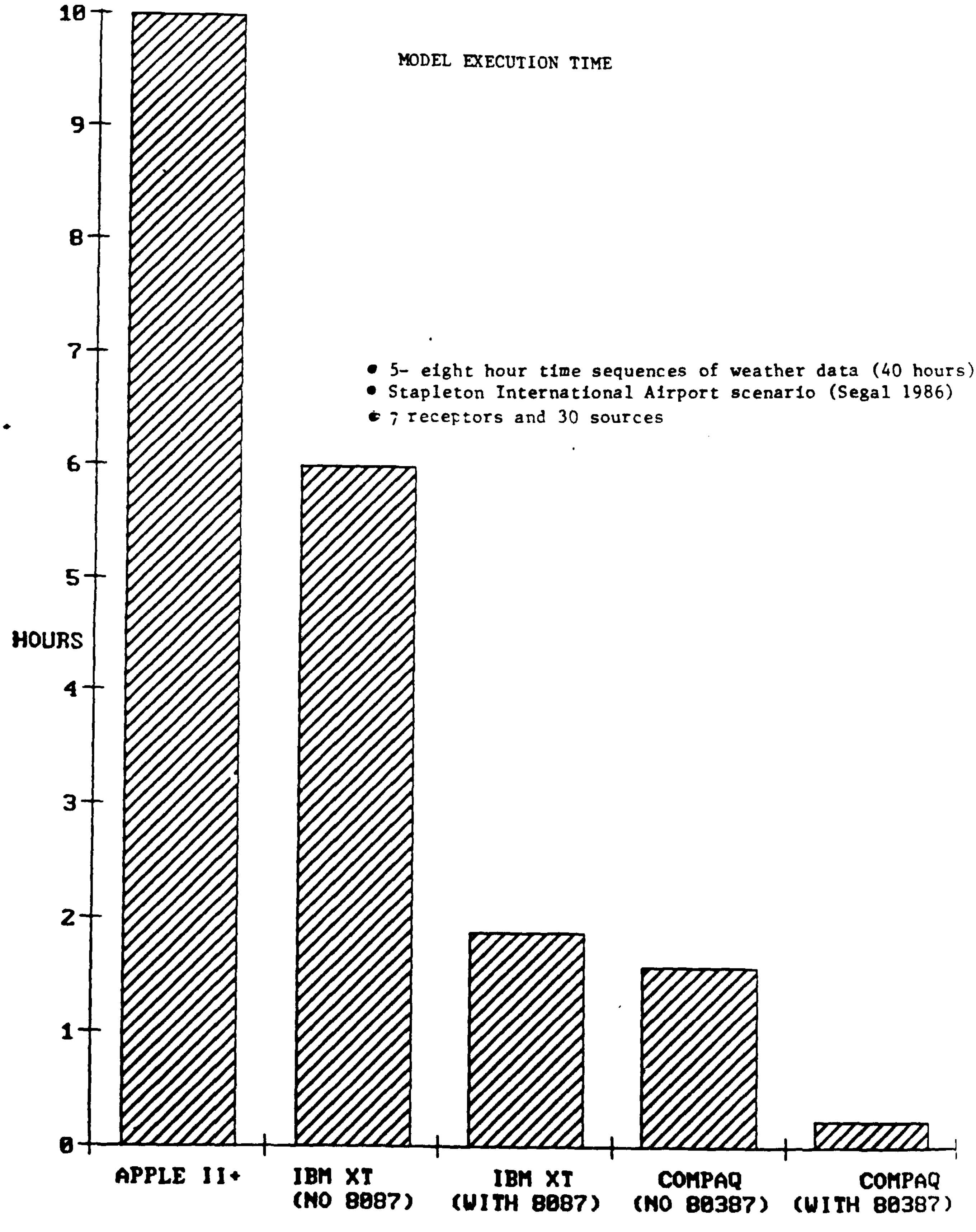
Figure 4 compares the speed at which the GIMM dispersion model calculates concentrations while using the following computers:

1. Apple II
2. IBM XT
3. IBM XT with math coprocessor
4. Compaq (a more modern computer using the 386 microprocessor)
5. Compaq with math coprocessor.

The GIMM run was for five 8-hour meteorological data sets (40 total hours) in which 30 sources and 7 receptors were processed. (It is reasonable to expect this amount of processing to assess conformance with the 8-hour Carbon Monoxide standard.)

The Compaq computer with the mathematics coprocessor took 13 minutes to complete this task. This test was run on May 13, 1988, with the computer technology and model development status as of that date.

The time to process more than 40 hours of meteorological data can be determined by extrapolating from the results shown in Figure 4.



MODEL EXECUTION TIME
• 5- eight hour time sequences of weather data (40 hours)
• Stapleton International Airport scenario (Segal 1986)
• 7 receptors and 30 sources
HOURS
10
9
8
7
6
5
4
3
2
1
0
APPLE II+
IBM XT (NO 8087)
IBM XT (WITH 8087)
COMPAQ (NO 80387)
COMPAQ (WITH 80387)

Figure 4

5 - COMPARISON OF EDMS WITH OTHER MODELS

5.1 GENERAL

The method for establishing the performance of both EMISSMOD and GIMM was to select representative airfield scenarios and process these scenarios through EMISSMOD and GIMM. These same scenarios were also processed through criteria models for comparison purposes.

5.2 EMISSION MODEL (EMISSMOD) COMPARISONS

EMISSMOD performance was determined by running EMISSMOD and a criteria model through the same Pease AFB scenario. Pease AFB was selected because it included all EDMS sources.

The criteria models for EMISSMOD comparison were:

1. Methods and constants provided in the EPA report, AP-42 (EPA, 1983)

2. American Petroleum Institute listings (API, 1978)

3. AQAM model results (Pease, 1977).

Every source type included in EDMS was evaluated and the results are listed in Appendix B.

Emission rate differences between EMISSMOD and criteria models ranged between +2 and -4 percent.

5.3 DISPERSION MODEL (GIMM) COMPARISONS

Using the Figure 5 scenario, the concentration differences between GIMM and criteria models ranged between +8 and -3 percent. The details of this comparison are described in Appendix B.

GIMM and criteria models were also compared for a Stapleton International Airport scenario. The associated report (Segal, 1986) showed that concentration differences between GIMM and the appropriate criteria model ranged between +7 and -3 percent.

Test Scenario

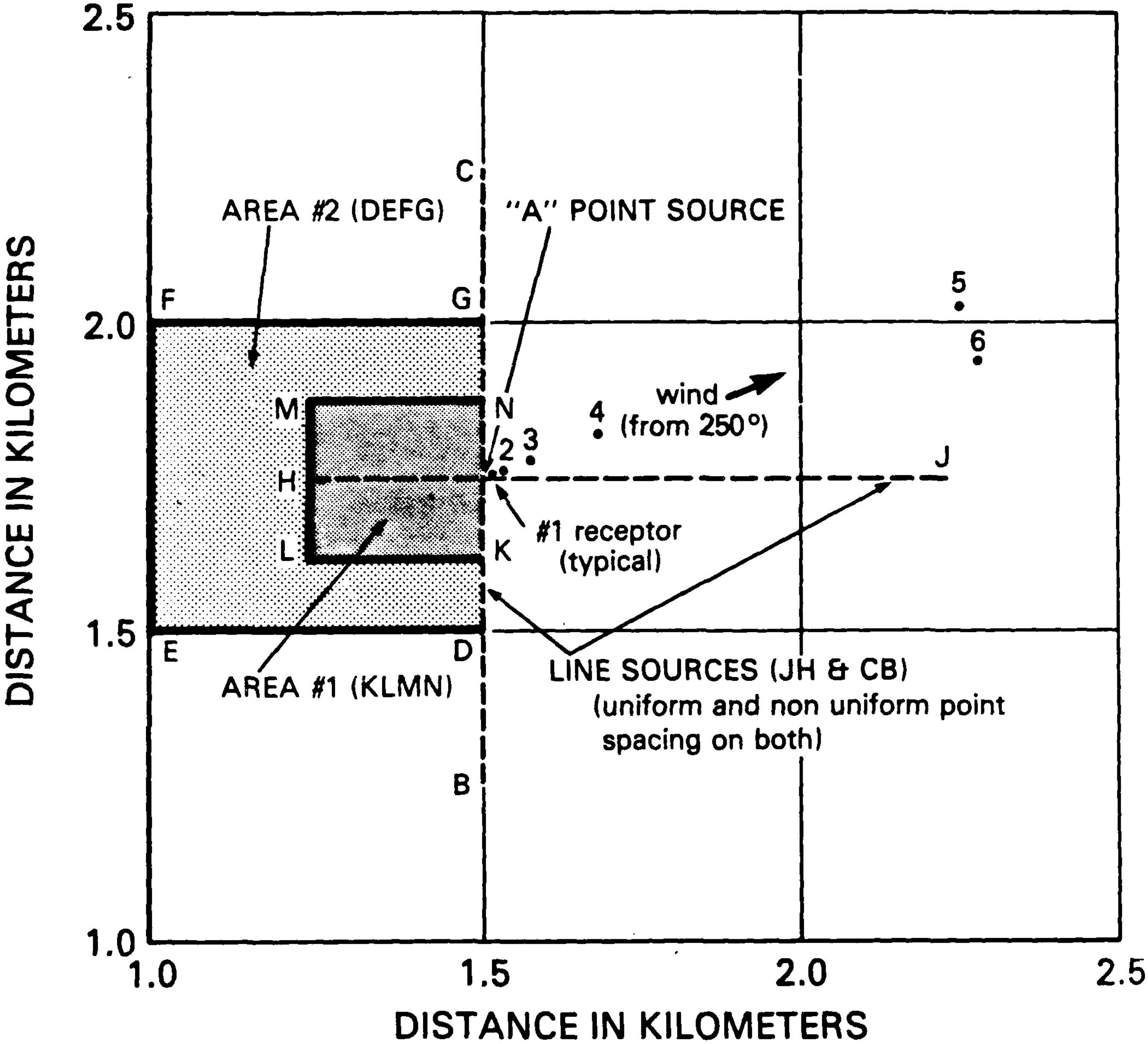


Figure 5

REFERENCES

Chock D. P.; General Motors Sulfate Dispersion Experiment: An Overview of the Wind, Temperature, and Concentration Fields; General Motors Reserach Laboratories; Warren, Mi 48090; GMR-2231; June 1977

Condor 1983; Condor 3 Database Management System - User's Guide; Condor Computer Corporation; P.O. Box 8318; Ann Arbor, MI.; 1983

EPA 1983; Compilation of Air Pollution Emission Factors - AP-42 Supplements 8 through 14; Environmental Protection Agency; Research Triangle Park, N.C.; June 1982 - May 1983

EPA 1985; Compilation of Air Pollutant Emission Factors - Volume II: Mobile Sources; Environmental Protection Agency; Ann Arbor, MI; September 1985

Pease 1977; Air Quality Assessment Scenario for Pease Air Force Base; United States Air Force Occupational and Environmental Health Laboratory; USAFOEHL/ECQ; Brooks AFB, Texas 78235 5501; 1977

Rau T. R. and Keenan M. T.; Suggestions for Improvement of the EPA-HIWAY Model; Journal of the Air Pollution Control Association; March 1980

Rote D. M. and Wangan L. E. ; A Generalized Air Quality Assessment Model for Air Force Operations"; Air Force Weapons Laboratory, Kirkland Air Force Base, New Mexico; USAF Report #AFWL -TR -74 -304, Feb. 1975

Segal H. M.; A Simplified Atmospheric Dispersion Model for Airport Use (User's Guide); Federal Aviation Administration, Washington, D. C.; Report FAA-EE-81-8; July 1981

Segal H. M.; Microcomputer Graphics in Atmospheric Dispersion Modeling; Journal of the Air Pollution Control Association; 23:6; June 1983

Turner D. B.; Workbook of Atmospheric Dispersion Estimates; U. S. Department of Health, Education, and Welfare; Environmental Health Service; Cincinnati, Ohio; Revised 1970

Yamartino R. J., Smith D. G., Bremer S. A., Heinold D., Lamich D., and Taylor B.; Impact of Aircraft Emissions on Air Quality in the Vicinity of Airports; Federal Aviation Administration; Washington, D. C. 20591; Report # FAA-EE-80-09A; July 1980

Wang I. T., Conley D. M., Rote D. M.; Airport Vicinity Air Pollution Model User's Guide"; Argonne National Laboratory, Argonne, Ill.; 1973

APPENDIX A

SUBMODELS OF THE GIMM DISPERSION MODEL

1. GIMM Submodels

GIMM incorporates the following four submodels:

I Facility Submodel

```
Powerplants----------points
Heating plants-------points
Fuel storage tanks---points
Training fires-------points
```

II Roadway Submodel-------points (spaced uniformly)

III Parking Lot Submodel---lines (finite)

IV Aircraft Submodel

```
Aircraft queues------points (spaced uniformly)
Aircraft takeoffs----points (spaced nonuniformly)
```

The facility submodel is broken down into four sub-submodels and the aircraft submodel is broken down into two sub-submodels. Concentrations for Facility and Aircraft submodels are reported as the sum of the concentrations from their sub-submodels.

While all four facility sub-submodels use the same dispersion algorithm the aircraft sub-submodels do not. Algorithm differences will be explained later on in this report.

For computational purposes, the line or area source geometry is broken down into the smaller elements noted above.

1.1 Facility Submodel

1.1.1 General

Powerplants, heating plants, training fires, and fuel storage tanks are modeled as facility sources. They all use the same point source dispersion algorithm from (Turner, 1970). However, each one reads a different emissions data file. Before addressing the method for calculating pollutant dispersion, the method for calculating plume rise is described:

1.1.2 Plume Rise Considerations

The following steps describe the plume rise calculations for all facility sources:

o Calculate volume flow rate:

```
WV = PI/4 * VS * D^2
```

where:

```
WV = Volume flow rate (m^3/sec)
PI = pi
VS = Stack flow speed (m/sec)
D  = Stack diameter (m)
```

o Calculate buoyant production:

```
F = g/PI * WV * (TS - T0)/T0
```

where:

```
g  = gravitational acceleration (m/sec^2)
     (3.12139 appears in code as g/PI
TS = Stack temperature (k)
T0 = Ambient temperature (k)
F  = Buoyant flux (m^4/s^3)
```

o Calculate plume rise.

a) For unstable or neutral conditions (PG <= 4):

```
XS = 14 * F^(5/8) (if F < 55 m^4/sec^3)
   = 34 * F^(2/5)   (if F >= 55 M^4/sec^3)
```

where:

XS = downwind displacement scale (m)

(Constants 14 and 34 are empirical; contain dimensions to convert XS to units of length in each case)

```
ZP = 1.6 * F^(1/3) * (3.5 * XS)^(2/3) * 1/U
```

where U is wind speed in meters/second, measured at the surface

ZP is plume rise in m

```
XF = 3.5 * XS (m)
```

XF = downwind displacement of plume in meters

b) For stable conditions:

i. First calculate the Brunt-Vaisala frequency:

N^2 = g * DT * 1/T0

where:

N = Brunt-Vaisala frequency (1/sec)
g = 9.8 m/sec^2
DT = potential temperature gradient, set to: .035 deg C/m

ii. Next, compute the plume rise in meters:

ZP = 2.6 * (F/(U * N^2))^(1/3)

-or-

ZP = 5.0 * (F^(1/4)) * (N^2)^(-3/8)
(whichever gives the smallest result)

iii. Under stable conditions, the plume will maintain its identity and drift downwind. The scale of the effective downwind displacement reflects the ratio of the windspeed to the Brunt-Vaisala frequency:

XF = PI * U/N

where:

XF = downwind displacement of source (m)

o Calculate effective source position:

ZD = H + ZP

XD = D - XF

XC = C

where:

ZD = effective source height (m)

H = stack or release height (m)

ZP = plume rise (m)

XD = downwind component of effective source-receptor displacement, origin at source (m)

D = source-receptor downwind distance (m)

XF = downwind plume displacement (m)

C = source-receptor crosswind distance (m)

XC = crosswind component of effective source-receptor displacement, origin at source (m)

1.1.3 Dispersion Considerations

Once the effective height and location of the point source plume have been determined, the next step is to calculate the concentration contribution at a receptor.

- o GIMM uses the downwind distance XD in a table lookup and linear interpolation routine to obtain the approximate Pasquill-Gifford sigma-y and sigma-z for the given stability class:

  SY = (TSY(I+1,PG) - TSY(I,PG)) * (XD/100 - I) + TSY(I)

  where:

  SY = sigma-y (m)

  I = INT(XD/100)

  TSY(I,PG) = tabulated horizontal standard deviation of plume concentration distribution at downwind distance of I * 100 m and Pasquill-Gifford stability class PG (m).

  SZ = (TSZ(I+1,PG) - TSZ(I,PG)) * (XD/100 - I) + TSZ(I)

  where:

  SZ = sigma-z (m)

  I = INT(XD/100)

  TSV(I,PG) = tabulated vertical standard deviation of plume concentration distribution at downwind distance of I * 100 m and Pasquill-Gifford stability class PG (m).

- o The next step is to scale the crosswind and vertical source-receptor distances by the appropriate sigmas:

  DS = C/SY (dimensionless crosswind distance)

  ZS = ZD/SZ (dimensionless vertical distance)

- o Obtain the relative horizontal and vertical concentration factors, (CH and CV) which will be dimensionless and range between 0 and 1:

  CH = (CR(I+1) - CR(I)) * (DS * 10 - I) + CR(I)

  CV = (CR(I+1) - CR(I)) * (ZS * 10 - I) + CR(I)

where:

I = int(DS * 10)

CR(I) = a table of the function
exp -((I / 10)^2)/2) (the Gaussian curve)
evaluated at I = 0, 1 ... 40

- Next, calculate the dimensionless relative concentration (sec/m^3) at this receptor:

CR = 1/(PI * SY * SZ * U) * CH * CV

- And the emission rate, Q(J) (gm/sec), for the hour of interest:

Q(J) = QF(J) * HF(HN) * DF(DN) * MF(MN)

where:

QF(J) = peak emission rate for species J in gm/sec. (The species indices correspond to the pollutants CO, HC, NOx, SOx, and particulates.)

Q(J) = emission rate for species J during hour HN, day DN, month MN, in gm/sec.

HF(HN) = temporal factor for hour HN, where HN goes from 0 to 23. The temporal factors give the fractional variation in activity through the hours of the day and run from 0 to 1.0.

DF(DN) = temporal factor for day DN, defined as above for DN from 1 to 7.

MF(MN) = temporal factor for month MN, defined as above for MN from 1 to 12.

- Compute the absolute concentration by multiplying the relative concentration at the receptor by Q(J) for all J pollutants.

- Finally, sum the concentration contributions for each pollutant, from each source, at each receptor.

1.2 Roadway Submodel

1.2.1 General

Roadway dispersion is modeled as a line source approximated by closely-spaced point sources. These points are evenly spaced along a line at an interval equal to the one-half sigma-y value at the receptor's downwind distance. Contributions from each point simulation of the roadway lines are computed and summed. Finally, the result is multiplied by the appropriate values of the peak emission rate--Q(J).

1.2.2 Point Source Spacing.

GIMM first obtains SG, the characteristic sigma-y value which is used for point spacing calculations. SG is determined in one of the following ways:

- If the receptor's crosswind coordinate falls between the crosswind components of the two endpoints, SG is based upon the y-intercept (D = 0) of the line source:

  D3 = D1 - C1 * S

  where:

  D3 = downwind distance used to compute SG.

  C1 = crosswind coordinate of leftmost end of line source.

  D1 = downwind coordinate of left side of line source.

  S = slope of line source.

```
                          (C2,D2)
                        /
                       /
                      /
                     / (0,D)
                    /:
                   / :
     line has     /  :
     slope = S   /   :D3
                /    :
               /     :
              /      :
     (C1,D1) /       :
                     R
               receptor (0,0)
```

- If the receptor is to the left of the leftmost endpoint,
  D3 = D1

- If the receptor is to the right of the rightmost endpoint, (C2,D2), then,

  D3 = D2

GIMM uses the table lookup routines to obtain a value of SG corresponding to the downwind distance D3, and reflecting any initial sigma-y which may have been established for the source. It then determines NP, the number of points to be used in the approximation:

NP = 2 * (C2-C1)/SG

(10 <= NP >= 200)

When approximating the line source, the point coordinates will be (CP,DP). The first point will be at the following position:

CP = C1 - .5 * CI
DP = D1 - .5 * DI

where:

CI = (C2 - C1)/NP
DI = (D2 - D1)/NP

Subsequent points are located by incrementing CP and DP by CI and DI until all NP have been used.

1.2.3 Line Clipping

Not all of a line segment will contribute to concentrations at a given receptor. In many cases, the computations may be speeded up by clipping both ends of a line segment so that the end portions which have an infinitesimal or no impact upon a receptor are eliminated. The following method is used:

- The first step is to define a region within which point sources can make contributions to the concentration measured at a receptor located at (0,0). Since the close-in Pasquill-Gifford curves for stabilities b - e all result in plume spreading at a less than linear rate with distance, it is conservative to approximate the envelope with the linear function, D < ABS(A * C).

```
                          (0,CT)
                ________________________________
      (C1,D1) \ (X1,Y1) | (X2,Y2) /     (C2,D2)
               \        |        /
                \       |       /
                 \      |      /
      slope=-A    \     |     /  slope=A
                   \    |    /
                    \   |   /
                     \  |  /
                      \ | /
                       \|/
                        R
                  receptor (0,0)
```

  where D = downwind component
  C = crosswind component
  A = slope of the plume envelope

- The equation of the line source is as follows:

  D = S * C + CT

  where:

  S = line source slope, given by (D2-D1)/(C2-C1), with C1, D1, C2, D2 being endpoints of the roadway line segment.

  CT = value of D corresponding to C = 0. It is calculated from CT = D1 - S * C1.

- Solving for (X1, Y1) and (X2, Y2), the endpoints of the new, clipped line segment:

  X1 = -CT / (AC + S); Y1 = -AC * X1
  X2 = CT / (AC - S); Y1 = AC * X1

- All that remains is to check whether either or both of the points (X1,Y1) and (X2,Y2) fall on the line segment between the original endpoints (C1,D1) and (C2,D2). If so, (C1,D1) and/or (C2,D2) are set equal to the corresponding clipped coordinate(s) and the emission rate of the line source reduced in order to maintain a constant line density.

1.2.4 Dispersion Table Modification

Studies by General Motors of actual dispersion from moving automobiles on a test track (Chock 1978) suggest a number of corrections and adjustments to the Gaussian plume algorithm. These primarily serve to account for the enhanced turbulence near the road due to the motion of the automobiles. The adjustments (Rao and Keenan, 1980) are as follows:

- Aerodynamic drag factor:

  UC = C * (U^.164) * (COS(A))^2

  where:

  UC = corrected windspeed (m/sec)
  U = ambient windspeed (m/sec)
  A = angle between road and x-axis
  C = 1.85 (empirical constant related to traffic speed)

  If UC < U then UC is set equal to U

- Initial dispersion of plume. The mechanical turbulence produced by moving automobiles causes considerable local mixing. GIMM models this with initial values of sigma-y and sigma-z at the source location. On the grounds that this initial turbulence will advect away from the road when the flow is primarily crossroad but stay over the road when the flow is down-road, this initial spreading is related to the magnitude of the crossroad wind component UX:

  SZ0 = 3.57m - .53*UX
  if SZ0 < 1.5 m then SZ = 1.5 m
  SY0 = 2.0 * SZ

  where:

  UX = U * COS(A)
  SZ0 = initial value of sigma-z
  SY0 = initial value of sigma-y

The Pasquill-Gifford dispersion curves have been modified to reflect the increased turbulence near the roadway. This section describes the values used for interpolation of the estimated sigma-y and sigma-z parameters.

For sigma-z:

The following sigma-z values are for up to 300 m downwind:

SZN(I,K) = A * (I)^B (I = 0, 100, 300 m)

where A, B and K are as follows:

PG class	A	B	K
B - C	110.62	.93198	1
D	86.49	.92332	2
E - F	61.14	.91465	3

These values are for beyond 300 m:

SZA(I,K) = SQR(SZN(300,K)^2 - TSZ(300,K)^2 + TSZ(I,J)^2)

I = 300, 400, ..., 10000 m

where:

TSZ(I,K) = table of Pasquill-Gifford sigma-z values, introduced in the "point sources" section.

SZN(I,K) = near field sigma-z values, for distances up to 300m, as defined above.

SZA(I,K) = table of adjusted sigma-z values.

K and J are defined as follows:

PG CLASS	K	J
B	1	2
C	1	3
D	2	4
E	3	5
F	3	6

For sigma-y:

SYN(I,K) = 465.1 * I * TAN(AP(I))

where:

AP(I) = C - D * LN(I /1000) (I = 0, 100, 200, 300 m)

where C, D and K are as follows:

PG CLASS	C	D	K
B - C	18.333	1.8096	1
D	14.333	1.7706	2
E - F	12.5	1.0857	3

beyond 300 m:

SYA(I,K) = SQR(SYN(300,J)^2 - TSY(300,K)^2 + TSY(I,J)^2)

I = 300, 400, ..., 10000 m

where:

TSY(I,K) = table of Pasquill-Gifford sigma-y values, introduced in the "point sources" section.

SYN(I,K) = near field sigma-y values, for distances up to 300m, as defined above.

SYA(I,K) = table of adjusted sigma-y values.

K and J are defined as above:

PG CLASS	K	J
B	1	2
C	1	3
D	2	4
E	3	5
F	3	6

Finally, the sigma values interpolated from the tables of SYA(I,K) and SZA(I,K) above are adjusted for the initial sigma values by squaring the sigma's for initial and downwind distances, summing these values, and extracting the square root of the sum. (This procedure shall be called "quadrature" in this report.)

```
SY = SQR(SYI^2 + SY0^2)
SZ = SQR(SZI^2 + SZ0^2)
```

where:

SY, SZ are the values of sigma-y and sigma-z used in the point source contribution.

SYI, SZI are the values of the sigmas interpolated from the tables SYA(I,K) and SZA(I,K).

1.3 Parking Lot Submodel

Parking lot emissions are modeled as area sources. The integration elements are line segments oriented normal to the direction of the wind. The calculation proceeds in the same manner as that for the point source, except that, instead of a table lookup for the Gaussian distribution curve:

```
CR(I) = exp -((I / 10)^2)/2)
        for I = 0 to 40
```

the table contains values for the area beneath the curve between zero and .0, .1, .2, .3, ... 4.0:

$$CR(I) = \int_{K=0}^{K=I/10} \exp -((K^2)/2)\ dK$$

for I = 0 to 40

The tabulated values may be linearly interpolated as before. All that has to be done is to evaluate the integral for each endpoint of the finite normal line source. Since this line is perpendicular to the wind, the downwind components of each endpoint will be the same. The calculation proceeds as follows:

- First, calculate sigma-y and sigma-z corresponding to the downwind source-receptor spacing. This is done using the same Pasquill-Gifford interpolation scheme outlined in the "POINT SOURCES" section. The modified sigmas used for road sources are not used here, the assumption being made that the parking lots will have less mechanical turbulence than the roads due to the slower speeds of the cars traveling in the parking lots.

- Next, scale the crosswind coordinates of the endpoints of the line segment by sigma-y. These scaled coordinates DS1 and DS2 are analogous to the scaled value DS in the "POINT SOURCES" section.

- The computation of relative concentration is also analogous to the one for point sources. The only difference is that there are now two interpolated table values, and they represent the areas under the normal distribution curve between 0 and each line segment endpoint. The contribution from the emissions between the two endpoints will be the absolute difference between the two values:

```
CN1 = (CR(I1+1) - CR(I1)) * (DS1 * 10 - I1) + CR(I1)
CN2 = (CR(I2+1) - CR(I2)) * (DS2 * 10 - I2) + CR(I2)
CN  = ABS( CN2 - CN1 )
```

where:

DS1 = crosswind coordinate, scaled by sigma-y, of one endpoint of the line segment

DS2 = crosswind coordinate, scaled by sigma-y, of the other endpoint of the line segment

I1 = int(DS1 * 10)

I2 = int(DS2 * 10)

CN1 = area under the Gaussian curve from 0 to DS1

CN2 = area under the Gaussian curve from 0 to DS2

CN = relative concentration at the receptor due to the line source emissions

1.4 Aircraft Submodel

1.4.1 General

Aircraft are small contributors to pollutant concentrations at airports (Yamartino, et al., 1980), while operating in the operational modes defined by the EPA. While concentrations would be expected to be small in all modes, the two modes producing the greater concentrations are takeoff and queueing. The aircraft path during these two modes is described as follows:

- The takeoff path extends from the downwind end of the runway to the point where it takes off. This source is treated as a nonuniform line.

- A queue path extends along the pre-takeoff taxiway up to the point where it intersects the downwind end of the runway. This source is treated as a uniform density line.

1.4.2 Aircraft Queue

The aircraft queue submodel is similar to the automobile road source submodel except that it does not incorporate the special sigma-y and sigma-z curves, as defined by (Rao, et al., 1980) at the close-in source-receptor distances.

1.4.3 Accelerating Aircraft.

The method described by (Segal 1981) has been incorporated into the model. Motion along the takeoff line is simulated by points whose spacing varies directly with the speed of the aircraft as it proceeds down the runway. GIMM assumes a 1 m/sec^2 acceleration. A time spacing of one point source every one-eighth second during the takeoff run is used to define point spacing. Consequently, the point spacing is as follows:

x = .5 * a * t^2

for a = 1 m/s, so that:

XP(T) = .5 * T^2 m/sec^2

where T = (0, .125, .250,......42.0)
(42 seconds is the EPA time in mode for takeoff)

Dispersion calculations for these nonuniform line sources (accelerating aircraft) are similar to those for uniform line sources (*queued aircraft*) in that the calculated concentrations from all points simulating the takeoff path are summed. Once all of the points have been accounted for, the relative concentration is scaled and multiplied by the adjusted emission rate Q(J) for an accelerating airplane.

APPENDIX B

MODEL COMPARISON RESULTS

Comparison of the EDMS With Other Models

1.1 General

The method for establishing the performance of both EMISSMOD and GIMM was to select representative airfield scenarios and process these scenarios through the EMISSMOD and GIMM models. These same scenarios were then processed through criteria models for comparison purposes.

1.2 Emission Model (EMISSMOD) Comparison

EMISSMOD performance was determined by running EMISSMOD and a criteria model through the same Pease AFB scenario. Pease AFB was selected because it included all sources listed in the EDMS.

The criteria documents for EMISSMOD comparison were:

1. Methods and constants provided in the EPA report, AP-42 (EPA 1980)

2. American Petroleum Institute listings (API 1978)

3. AQAM model results (Pease 1977). (The Pease AFB scenario was processed by AQAM in 1977.)

Every source type included in EDMS was evaluated and the results are listed in tables 1-4.

Emission rate differences between EMISSMOD and criteria model results ranged between +4 and -3 percent.

TABLE B1

MODEL COMPARISON RESULTS

TRAINING FIRES

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	2.95E 07	1.69E 07	2.19E 05	5.27E 04	6.74E 06
AQAM	2.95E 07	1.69E 07	2.19E 05	5.27E 04	6.74E 06

TANK FARMS

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	0.00E 00	1.29E 06	0.00E 00	0.00E 00	0.00E 00
API	0.00E 00	1.27E 06	0.00E 00	0.00E 00	0.00E 00

NATURAL GAS HEATING PLANT

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	1.09E 06	2.14E 05	5.44E 06	0.00E 00	1.63E 05
AP-42	1.09E 06	2.14E 05	5.44E 06	0.00E 00	1.63E 05

FUEL OIL HEATING PLANT

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	1.23E 06	8.63E 05	1.78E 07	5.15E 07	6.78E 06
AP-42	1.23E 06	8.63E 05	1.78E 07	5.15E 07	6.78E 06

TABLE B2

MODEL COMPARISON RESULTS (CONT.)

INCINERATOR

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	2.03E 08	1.63E 07	2.44E 07	1.22E 07	6.50E 05
AQAM	2.03E 08	1.63E 07	2.44E 07	1.22E 07	6.50E 05

AUTOMOBILES ON ROADWAY -- 100% COLD STARTED AT 5 MPH

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	5.76E 06	3.95E 05	3.00E 04	9.00E 01	3.11E 03
AP-42	5.76E 06	3.95E 05	3.00E 04	9.00E 01	3.11E 03

AUTOMOBILES ON ROADWAY -- STABILIZED AT 30 MPH

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	6.93E 04	7.97E 03	9.47E 03	4.49E 01	1.56E 03
AP-42	6.93E 04	7.97E 03	9.47E 03	4.49E 01	1.56E 03

AUTOMOBILES ON ROADWAY -- 20% COLD START AT 30 MPH

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	1.86E 05	1.60E 04	1.06E 04	4.58E 01	1.59E 03
AP-42	1.86E 05	1.60E 04	1.06E 04	4.58E 01	1.59E 03

TABLE B3

MODEL COMPARISON RESULTS (CONT.)

AUTOMOBILES ON ROADWAY -- 20% COLD START AT 30 MPH (SHORTER ROAD)

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	1.90E 04	1.63E 03	1.08E 03	4.67E 00	1.62E 02
AP-42	1.89E 04	1.66E 03	1.09E 03	4.58E 00	1.61E 02

AUTOMOBILE PARKING LOT

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	4.45E 03	1.70E 02	1.41E 01	4.94E -1	1.71E 00
AP-42	4.45E 03	1.70E 02	1.41E 01	4.94E -1	1.61E 00

AIRPLANES -- "TRANSENT" (AQAM GENERIC TRANSIENT)

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	8.79E 07	1.74E 07	1.41E 07	1.49E 07	3.11E 06
AQAM	8.77E 07	2.57E 07	1.41E 07	1.41E 07	3.06E 06
(FUELING)		-8.34E 06			
AQAM	8.77E 07	1.74E 07	1.41E 07	1.41E 07	3.06E 06

AIRPLANES -- B727

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	1.37E 08	3.69E 07	5.43E 07	5.90E 06	1.95E 06
AP-42	9.26E 07	2.22E 07	4.91E 07	5.40E 06	1.94E 06
GSE	4.02E 07	8.94E 06	2.85E 06	-	-
APU	4.86E 06	1.21E 06	3.69E 06	6.46E 05	-
SHUTDOWN	-	4.23E 06	-	-	-
TOTAL	1.38E 08	3.66E 07	5.56E 07	5.94E 06	1.94E 06

TABLE B4

MODEL COMPARISON RESULTS (CONT.)

AIRPLANES--B737

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	7.93E 07	2.09E 07	3.66E 07	4.19E 06	1.29E 06
AP-42	6.18E 07	1.49E 07	3.27E 07	3.61E 06	1.28E 06
GSE	1.21E 07	2.69E 06	8.07E 05	-	-
APU	5.40E 06	1.50E 06	3.33E 06	5.88E 05	-
SHUTDOWN	-	2.83E 06	-	-	-
TOTAL	7.92E 07	2.19E 07	3.68E 07	4.20E 06	1.28E 06

AIRPLANES--DH-6

MODEL	CO	HC	NOx	SOx	Particulates
EDMS	1.18E 07	8.60E 06	1.35E 06	2.76E 05	-
AP-42	1.14E 07	8.60E 06	1.35E 06	2.92E 05	-

1.3 Dispersion Model Comparisons

The source-receptor scenario for the GIMM evaluation is shown in figure 5. These sources were processed through both GIMM and either HIWAY2 or PAL. The later two models acted as criteria models for evaluating GIMM performance. The results are listed in tables 5 through 8.

Concentration differences between GIMM and the criteria models for all sources and stability classes tested ranged between +8 and -3 percent.

TABLE B5

MODEL COMPARISON RESULTS

ACCELERATING AIRCRAFT

Wind Direction = 250°
Wind Velocity = 2m/sec
Emission Rate = 2.64 grams/sec

P/G Stability Class	Receptor #	Concentration (grams/m^3) GIMM	PAL	Percent Difference GIMM/PAL (%)
	1	2.71E-5	2.79E-5	-3
	2	2.70E-5	2.67E-5	+1
B	3	1.73E-5	1.75E-5	-1
	4	1.09E-5	1.11E-5	-2
	5	3.04E-6	3.05E-6	0
	6	5.38E-6	5.39E-6	0
	1	3.17E-5	3.10E-6	+2
	2	3.57E-5	3.40E-5	+5
E	3	3.30E-5	3.17E-5	+4
	4	2.78E-5	2.68E-5	+3
	5	1.41E-5	1.39E-5	+1
	6	2.04E-5	2.00E-5	+1

TABLE B6

MODEL COMPARISON RESULTS

AREA SOURCE

Wind Direction = 250^0
Wind Velocity = 2m/sec
Emission Rates:
(Area 1) = 5.776×10^{-4} grams/m^2-sec
(Area 2) = 1.444×10^{-5} grams/m^2-sec

P/G Stability Class	Receptor #	Concentration (grams/m^3)		Percent Difference
		GIMM	PAL	GIMM/PAL (%)
	1	7.93E-3	7.68E-3	+3
	2	6.44E-3	6.32E-3	+2
B	3	3.74E-3	3.74E-3	0
	4	2.41E-3	2.42E-3	0
	5	5.67E-4	5.66E-4	0
	6	4.10E-4	4.09E-4	0
	1	2.11E-2	1.98E-2	+7
	2	1.92E-2	1.81E-2	+6
E	3	1.29E-2	1.27E-2	+2
	4	9.43E-3	9.36E-3	+1
	5	4.08E-3	4.07E-3	0
	6	2.03E-3	2.02E-3	0

TABLE B7

MODEL COMPARISON RESULTS

POINT SOURCE

Wind Direction = 250^{0}
Wind Velocity = 2m/sec
Emission Rate = 0.75 grams/sec

P/G Stability Class	Receptor #	Concentration (grams/m^3) GIMM	Concentration (grams/m^3) PAL	Percent Difference GIMM/PAL (%)
	1	1.77E-3	1.78E-3	0
	2	1.65E-3	1.62E-3	+1
B	3	4.70E-4	4.79E-4	-2
	4	2.12E-4	2.15E-4	-1
	5	2.09E-4	2.10E-5	0
	6	1.54E-4	1.55E-5	-1
	1	7.21E-4	6.75E-4	+7
	2	1.42E-3	1.32E-3	+8
E	3	1.63E-3	1.64E-3	0
	4	9.16E-4	9.28E-4	-1
	5	1.99E-4	2.00E-4	0
	6	1.51E-4	1.49E-4	+1

TABLE B8

MODEL COMPARISON RESULTS

LINE SOURCE

Wind Direction = 250^0
Wind Velocity = 2m/sec
Emission Rate = 0.27 grams/sec

P/G Stability Class	Receptor #	Concentration ($grams/m^3$) GIMM	Concentration ($grams/m^3$) HIWAY 2	Percent Difference GIMM/HIWAY 2 (%)
B	1	9.73E-2	9.25E-2	+5
	2	8.32E-2	7.86E-2	+6
	3	3.10E-2	3.08E-2	+1
	4	1.59E-2	1.58E-2	+1
	5	4.01E-3	3.86E-3	+4
	6	6.23E-3	5.94E-3	+5
E	1	1.11E-1	1.07E-1	+4
	2	1.09E-1	1.04E-1	+5
	3	5.33E-2	5.29E-2	+1
	4	2.90E-2	2.89E-2	0
	5	1.54E-2	1.54E-2	0
	6	1.81E-2	1.81E-2	0

APPENDIX C

SOURCE CODE DESCRIPTION

TABLE OF CONTENTS

VARIABLE LIST

NAME	TYPE	DESCRIPTION
A	COUNTER	COUNTS OFF PT$ IN SUBSTRINGS OF 8 CHARACTERS EACH
A	COUNTER	FS(a) -- COUNTER FOR 7 DAYS OF WEEK
A$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Z (TABLE LOOKUP)
AN$	STRING	REPLY CHARACTER ENTERED BY USER
B$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Y (TABLE LOOKUP)
C	REAL	CROSSWIND CONCENTRATION FACTOR
C	INTEGER	ASSIGNMENT OF LEFT 2 CHARACTERS OF INPUTTED YEAR (CENTURY)
C1	REAL	CROSSWIND COORDINATE OF LINE ENDPOINT
C2	REAL	CROSSWIND COORDINATE OF LINE ENDPOINT
C3	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
C4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
chr$15	string	control character for prowriter printer (condensed mode)
CHR$27	STRING	CONTROL CHARACTER FOR PROWRITER PRINTER LINE LENGTH
CHR$81	STRING	CONTROL STRING FOR PROWRITER PRINTER LINE LENGTH
D	INTEGER	ASSIGNMENT OF RIGHT 2 CHARACTERS OF INPUTTED YEAR(100 YRS)
D	REAL	DOWNWIND CONCENTRATION FACTOR
D%(D1)	INTEGER	D%(7) TEMP FACTOR - WEEKDAY (30% ACT = .3 * 10)
D1	REAL	DOWNWIND COORDINATE OF LINE ENDPOINT
D2	REAL	DOWNWIND COMPONENT OF LINE ENDPOINT
D3	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
D4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
DXY	REAL	VIRTUAL POINT INTERPOLATION VALUE
DXZ	REAL	VIRTUAL POINT INTERPOLATION VALUE
DY	REAL	CHANGE IN SIGMA Y BETWEEN INDICES I AND I+1
DZ	REAL	CHANGE IN SIGMA Z BETWEEN INDICES I AND I+1
F	REAL	BUOYANCY FLUX

VARIABLE LIST

NAME	TYPE	DESCRIPTION
F$	STRING	F$(6) SUBSCRIPTED VAR -- READS DAY OF WEEK DATA STRING
FC	INTEGER	DAY OF MONTH (CONVERTS DAY OF WEEK TO ONE OF 5 WEEKS IN MO
FM$	STRING	FM$(A) STRINGS USED TO SET UP REPORT FORMAT (12"A" VALUES)
FM$(A)	STRING	TEXT STRING FOR PRINTED OUTPUT REPORT
G	REAL	NORMALIZED CONCENTRATION
GP	REAL	g/PI = 3.12139
H	INTEGER	HOUR OF DAY (REDUNDENT WITH H1)
H	REAL	EFFECTIVE SOURCE HEIGHT
H%(H1)	INTEGER	H%(24) TEMP FACTOR - HOUR OF DAY (50% ACT = .5 * 10)
H1	INTEGER	HOUR OF DAY
HF	INTEGER	HOURS IN YEAR CONSTANT--8760
I	COUNTER	Y/SIGMA Y RATIO * 10 -- 40 TABLE LOOKUP POINTS (4 SIGMA)
I	COUNTER	DISTANCE INCREMENTAL INDEX FOR P/G TABLE LOOKUP (101 DIST)
IN$	string	8 CHARACTER READ OF TEMPROAD --IN$(A) OR IN$ (K)-K &A SAME
IN$	STRING	INPUT VARIED LENGTH STRINGS OF ROADFILE--C/D,%CLD,SIG,TEMP
IXY	INTEGER	VIRTUAL POINT INDEX FOR TABLE INTERPOLATION
IXZ	INTEGER	TABLE INDEX FOR VIRTUAL POINT INTERPOLATION
J	COUNTER	P/G CLASS INCREMENTAL INDEX 3 CLASSES (TABLE LOOKUP)
J1	COUNTER	COUNTER
K	INTEGER	GENERAL COUNTER
KF	INTEGER	Y - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KG	INTEGER	X - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KH	INTEGER	Y - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KI	INTEGER	X - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS

VARIABLE LIST

NAME	TYPE	DESCRIPTION
M	INTEGER	ASSIGNED INTEGER VALUE OF STRING MONTH i.e."january"(1ft3)
M$	STRING	USER INPUT MONTH STRING
M%(M1)	INTEGER	M%(12) TEMPORAL FACTOR - MONTH (40% ACTIVITY =.4 * 10)
M1	INTEGER	MONTH --REDUNDENT WITH "M"
N	REAL	BRUNT-VAISALA FREQUENCY
ND(I)	REAL	ND(I) GAUSSIAN EXPONENTIAL VALUE (0 - 1) TABLE LOOKUP
P2	REAL	STABILITY INDEX FOR TABLE LOOKUP
PC(I)	REAL	POLLUTANT CONCENTRATION FOR SPECIES I
PG	INTEGER	PASQUILL/GIFFORD STABILITY CLASS
PI	REAL	PI = 3.14159
Q(P)	REAL	Q(5) INTERPOLATED EMISSION RATE -- GM/S (5 POLLUTANTS)
QX	REAL	DUMMY VARIABLE USED IN STRING CLEARING
RR	COUNTER	COUNTER FOR RECEPTORS
ST	REAL	STACK TEMPERATURE
STDIA	REAL	STACK DIAMETER
TE	real	TE(5) - TOTAL EMISSIONS - SUM OF "Q" (5 POLLUTANTS)
TEMP	REAL	USER INPUT TEMPERATURE
TEMP	REAL	AMBIENT TEMPERATURE
TF	INTEGER	EQUIVALENT HOURS OF FULL POWER OPEPATION
TH	REAL	WIND ANGLE IN RADIENS
TS	REAL	LAPSE RATE
TS	REAL	POTENTIAL TEMPERATURE SLOPE
VS	REAL	STACK FLOW SPEED
W%(W1)	INTEGER	W%(5) temp factor - monthweek (30% activity = .3 * 10)
W1	INTEGER	WEEK

VARIABLE LIST

NAME	TYPE	DESCRIPTION
WS	REAL	USER INPUT WIND SPEED
WS	REAL	WIND SPEED
WV	REAL	VOLUME FLOW RATE
X$	STRING	READ IN "x" COORD STRING FOR RECEPTOR - 8 CHAR LONG
XF	REAL	DOWNWIND PLUME DISPLACEMENT
XR	REAL	"X" COORDINATE OF RECEPTOR
Y$	STRING	READ IN "Y" COORD OF RECEPTOR RANDOM ACCESS 8 CHAR LONG
Y0	REAL	INITIAL PLUME SIGMA Y
Y1	REAL	"Y" COORDINATE OF ROAD ENDPOINT

VARIABLE LIST

NAME	TYPE	DESCRIPTION
Y2	REAL	"Y" COORDINATE OF OTHER ENDPOINT OF ROAD
YR	REAL	"Y" COORDINATE OF RECEPTOR
YR$	STRING	USER INPUT OF "YEAR"
YT	REAL	TABULATED SIGMA Y VALUES ASSIGNED FOR P/G TABLE LOOKUP
Z0	REAL	INITIAL PLUME SIGMA Z
ZP	REAL	PLUME RISE
ZQ	INTEGER	BOOLEAN CONTROL (0 or 1) IF INPUT ERROR IN DATE, WEATHER
ZT	REAL	TABULATED SIGMA Z VALUES ASSIGNED FOR P/G TABLE LOOKUP

```
*****************************************************************************
*                                                                           *
*                       POINT SOURCE DISPERSION MODEL                       *
*                                                                           *
*****************************************************************************


10  DIM IN$(84), M%(12), W%(5), F1(15), D%(7), H%(24), FM$(18), E(5)
12  DIM FA$(4), F$(6), PC(5), Q(5)
15  DIM ND(40), TC(5, 5), ZT(100, 3), YT(100, 3), TE(5), GM$(10)
19  P1 = 0
20  GOSUB 15000 ' DEFINE FAC$ AS POWER/HEAT PLANTS
22  HOURN = 0
24  INPUT "SCREENING - 'S'   OR   REFINED - 'R'"; BAT$
26  GOSUB 16070: IF BAT$ <> "S" AND BAT$ <> "R" THEN 24
28  IF BAT$ = "I" THEN 30 ELSE OPEN "I", #6, "WEATHER.DAT"
30  OPEN "I", #1, FACF$
40  OPEN "I", #3, FAC$
60  FOR A = 1 TO 4: READ FA$(A): NEXT A
65  DATA "POWER/HEAT","INCINERATOR","TRAINING FIRE","FUEL FACILITY"
80  LPRINT CHR$(27); CHR$(81)
90  WIDTH "LPT1:", 132
93  TWR = 1
95  GOSUB 1470
96  P = 0
99  J1 = 0
115  GOSUB 2830 ' SET UP FOR INTERP OF GAUSS CURVE
116  GOSUB 2960 ' READ INTERPOLATION TABLE FOR SIGMAS
118  GOSUB 1590 ' INPUT TIME AND WEATHER
119  GOSUB 1190 ' ZELLERS CONGRUENCE LAW
120  IF HOURN = 0 THEN GOSUB 5100 ' SETUP AND PRINT REPORT
122  GOSUB 3900 ' INPUT RECEPTOR LOCATION
125  GOTO 1020 ' READ SOURCE
126  GOSUB 1800 ' TEMPORAL FACTOR SUBROUTINE
130  GOSUB 1500 ' TEMPERATURE INTERPOLATION
135  GOSUB 15300 'PLUME RISE CALCULATION
140  GOSUB 2600 ' CALCULATE DISPERSION
150  GOTO 125
160  END
```

```
***************************************************************************
*                                                                         *
*   NAME:              FILEREAD                                           *
*   FUNCTION:          READS IN FACILITY FILES                            *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                               *
*                                                                         *
***************************************************************************

1051  H = VAL(IN$(22)): Z0 = 1.5
1052  Y0 = 3
1055  FOR K = 1 TO 7: LET D%(K) = 10 * VAL(IN$(K + 30)): NEXT K
1060  FOR K = 1 TO 5: LET W%(K) = 10 * VAL(IN$(K + 37)): NEXT K
1065  FOR K = 1 TO 12: LET M%(K) = 10 * VAL(IN$(K + 42)): NEXT K
1070  FOR K = 1 TO 24: LET H%(K) = 10 * VAL(IN$(K + 54)): NEXT K
1075  FOR K = 1 TO 5: LET E(K) = 10 * VAL(IN$(K + 78)): NEXT K
1080  QX = FRE(0)
1090  EX = 0
1100  INPUT #3, IN$(1)
1110  FOR K = 1 TO LEN(IN$(1)) STEP 8: EX = EX + 1: E(EX) = VAL(MID$(IN$(1),
      K, 8)): NEXT K
1153  X = VAL(IN$(19)): Y = VAL(IN$(20))
1160  STACKHGT = VAL(IN$(28)): STACKDIA = VAL(IN$(29)): BUILDHGT = VAL(IN$(25))
1165  STACTEMP = VAL(IN$(26)): STACFLOW = VAL(IN$(27)): IF STACFLOW < 5 THEN
      STACFLOW = 5
1175  GOTO 126

***************************************************************************
*                                                                         *
*   NAME:              DAY                                                *
*   FUNCTION:          USES ZELLER'S CONGRUENCE LAW TO DETERMINE DAY OF WEEK *
*                      FROM DATE                                          *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                               *
*   CALLS:             NO OTHER ROUTINES                                  *
*                                                                         *
***************************************************************************

1190  CLS
1200  C = VAL(LEFT$(YR$, 2)): D = VAL(RIGHT$(YR$, 2))
1210  GOSUB 1337
1220  M1 = INT(M): W1 = INT(FC / 7) + 1: H1 = INT(HR)
1230  M = M - 2: IF M < 1 THEN D = D - 1: M = M + 12
1240  IF D < 0 THEN D = 99 + D: C = C - 1
1250  FK = (INT(2.6 * M - .2 + .00001) + FC + D + INT(D / 4 + .00001) +
      INT(C / 4 + .00001) - (2 * C))
1260  Y = FK / 7
1270  Z = ABS(Y - INT(Y + .00001))      'DECIMAL REMAINDER
1280  DT = INT(7 * Z + .00001)   'INTEGER REMAINDER
1290  DT = DT + 1
1330  RETURN
```

```
************************************************************************
*
*   NAME:             MONTH
*   FUNCTION:         ASSIGNS MONTH INDEX M FOR SUBROUTINE DAY
*   CALLED FROM:      MAIN -- MAIN PROGRAM
*   CALLS:            NO OTHER ROUTINES
*
************************************************************************

1337  M = 13
1340  IF LEFT$(M$, 3) = "JAN" THEN M = 1
1350  IF LEFT$(M$, 3) = "FEB" THEN M = 2
1360  IF LEFT$(M$, 3) = "MAR" THEN M = 3

1370  IF LEFT$(M$, 3) = "APR" THEN M = 4
1380  IF LEFT$(M$, 3) = "MAY" THEN M = 5
1390  IF LEFT$(M$, 3) = "JUN" THEN M = 6
1400  IF LEFT$(M$, 3) = "JUL" THEN M = 7
1410  IF LEFT$(M$, 3) = "AUG" THEN M = 8
1420  IF LEFT$(M$, 3) = "SEP" THEN M = 9
1430  IF LEFT$(M$, 3) = "OCT" THEN M = 10
1440  IF LEFT$(M$, 3) = "NOV" THEN M = 11
1450  IF LEFT$(M$, 3) = "DEC" THEN M = 12
1460  RETURN
************************************************************************
*
*   NAME:             WEEK
*   FUNCTION:         SETS UP TABLE OF NAMES OF DAYS
*   CALLED FROM:      MAIN -- MAIN PROGRAM
*   CALLS:            NO OTHER ROUTINES
*
************************************************************************

1470  FOR A = 0 TO 6: READ F$(A): NEXT A
1480  DATA "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRIDAY",
      "SATURDAY"
1490  RETURN

************************************************************************
*
*   NAME:             EMISSIONS
*   FUNCTION:         PUTS POINT EMISSIONS IN CORRECT UNITS
*   CALLED FROM:      MAIN -- MAIN PROGRAM
*   CALLS:            NO OTHER ROUTINES
*
************************************************************************

1540  FOR P = 1 TO 5
1550  Q(P) = E(P) / 60 * TF
1560  NEXT P
1580  RETURN
```

```
****************************************************************************
*                                                                          *
*   NAME:            INPUT                                                 *
*   FUNCTION:        INPUT OF METEROLOGY AND DATE FROM KEYBOARD            *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                  *
*   CALLS:           CONDITIONALLY CALLS THE FILE INPUT VERSION OF THIS    *
*                    ROUTINE                                               *
*                                                                          *
****************************************************************************

1590  CLS : IF BAT$ = "R" THEN 2000
1593  LOCATE 5, 10: INPUT "YEAR                        = "; YR$: IF ZQ > 0
      THEN RETURN
1594  LOCATE 7, 10: INPUT "MONTH                       = "; M$
1596  GOSUB 16000: GOSUB 1337: IF ZQ > 0 THEN RETURN
1598  IF M = 13 THEN GOTO 1594 ' CHECK MONTH IS IN BOUNDS
1600  LOCATE 9, 10: INPUT "DAY OF MONTH                = "; FC: IF ZQ > 0
      THEN RETURN
1605  DA = FC
1610  LOCATE 11, 10: INPUT "HOUR OF DAY (1-24)         = "; HR: IF ZQ > 0
      THEN RETURN
1620  LOCATE 13, 10: INPUT "TEMPERATURE (DEG F)        = "; TEMP: IF ZQ > 0
      THEN RETURN
1622  IF TEMP > 100 THEN TEMP = 100
1624  IF TEMP < 0 THEN TEMP = 0
1630  LOCATE 15, 10: INPUT "WIND SPEED (M/S)           = "; WS:V=WS:
      IF V < 1 THEN V = 1
1631  IF ZQ > 0 THEN RETURN
1635  LOCATE 17, 10: INPUT "WIND DIRECTION (DEG N) = "; W
1636  IF ZQ > 0 THEN RETURN
1640  LOCATE 19, 10: INPUT "P/G CLASS (2-5)            = "; PG: IF ZQ > 0
      THEN RETURN
1650  CLS
1660  ZQ = 0
1665  PRINT "2.MONTH = "; M$
1670  PRINT "3.DAY OF MONTH = "; FC
1675  PRINT "4.HOUR OF DAY = "; HR
1680  PRINT "5.TEMPERATURE = "; TEMP
1685  PRINT "6.WIND SPEED = "; WS
1690  PRINT "7.WIND DIRECTION = "; W
1695  PRINT "8.PG CLASS = "; PG
1702  INPUT "ARE THESE CORRECT (Y OR N)"; AN$
1703  IF AN$ <> "N" THEN RETURN
1704  ZQ = ZQ + 1
1705  INPUT "CHANGE WHICH VALUE"; AN
1706  CLS
1710  ON AN GOSUB 1593, 1594, 1600, 1610, 1620, 1630, 1635, 1640
1711  ZQ = 0
1715  GOTO 1650
```

```
***************************************************************************
*                                                                         *
*   NAME:           DUTY                                                  *
*   FUNCTION:       CALCULATE THE CORRECT TEMPORAL FACTOR                 *
*   CALLED FROM:    MAIN -- MAIN PROGRAM                                  *
*   CALLS:          NO OTHER ROUTINES                                     *
*                                                                         *
***************************************************************************

1800  TF = M%(M1) * W%(W1) * D%(DT)
1810  TF = TF / 10000
1820  TF = TF * H%(H1+1)
1830  RETURN

***************************************************************************
*                                                                         *
*   NAME:           WEATHER                                               *
*   FUNCTION:       FILE VERSION OF INPUT ROUTINE                         *
*   CALLED FROM:    INPUT -- SUBROUTINE                                   *
*   CALLS:          NO OTHER ROUTINES                                     *
*                                                                         *
***************************************************************************

2000  IF EOF(6) THEN 16500
2010  INPUT #6, ENP$
2015  IF ENP$="" THEN GOTO 2000
2020  YR$ = MID$(ENP$, 1, 4): YR = VAL(YR$)
2030  M$ = MID$(ENP$, 5, 3): GOSUB 1337
2040  FC = VAL(MID$(ENP$, 8, 2))
2050  HR = VAL(MID$(ENP$, 10, 2))
2060  TEMP = VAL(MID$(ENP$, 12, 3))
2070  WS = VAL(MID$(ENP$, 15, 2)) * .514668:V=WS:IF V<1 THEN V=1
2080  W = VAL(MID$(ENP$, 17, 3)
2090  PG = VAL(RIGHT$(ENP$, 1))
2100  IF WS < 1 THEN 2000
2110  IF PG < 3 THEN PG = 3
2120  RETURN
```

```
**********************************************************************
*                                                                    *
*   NAME:            DISPERSION                                      *
*   FUNCTION:        CONTROL LOOP FOR ALL DISPERSION CALCULATIONS    *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*   CALLS:           DRAW, DUTY, ROTATE, LENGTH, DELTA, CONC, PRINT  *
*                                                                    *
**********************************************************************

2600  C1 = VAL(IN$(19)): D1 = VAL(IN$(20)): C2 = C1: D2 = D1
2605  GOSUB 4500' DRAW SOURCE
2620  RX = XR: RY = YR' RECEPTOR X AND Y LOCATION
2705  GOSUB 3700' ROTATE/TRANSLATE
2740  IF PG > 5 THEN PG = 5
2750  IF PG < 2 THEN PG = 2
2764  GOSUB 3550' GET CONCENTRATION
2765  GOSUB 12000' PRINT RESULTS
2766  RETURN

**********************************************************************
*                                                                    *
*   NAME:            GAUSSTABLE                                      *
*   FUNCTION:        CALCULATE THE GAUSSIAN PLUME TABLE              *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*   CALLS:           NO OTHER ROUTINES                               *
*                                                                    *
**********************************************************************

2840  FOR I = 0 TO 40
2850  ND(I) = EXP(-.005 * I * I)
2860  NEXT I
2870  RETURN

**********************************************************************
*                                                                    *
*   NAME:            NORMAL                                          *
*   FUNCTION:        CALCULATE THE NORMAL DIST CURVE                 *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*   CALLS:           NO OTHER ROUTINES                               *
*                                                                    *
**********************************************************************

2890  IF ABS(X) > 4 THEN FX = 0: GOTO 2950
2900  X = ABS(X) * 10
2910  IX = INT(X)
2920  DX = (X - IX)
2930  FA = ND(IX) - ND(IX + 1)
2940  FX = ND(IX) - FA * DX
2950  RETURN
```

```
*************************************************************************
*                                                                       *
*   NAME:             SIGMATABLE                                        *
*   FUNCTION:         READS IN SIGMA TABLE INFORMATION FROM FILE        *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                              *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

2960  OPEN "SIGMA" FOR INPUT AS #2
2980  FOR J = 0 TO 3
2990  FOR I = 0 TO 100
3000  INPUT #2, A$, B$: ZT(I, J) = VAL(A$): YT(I, J) = VAL(B$)
3010  NEXT I
3020  NEXT J
3030  CLOSE #2
3040  RETURN

*************************************************************************
*                                                                       *
*   NAME:             SIGMA                                             *
*   FUNCTION:         INTERPOLATION FOR SIGMAS                          *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE               *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

3060  IF X > 9999 THEN X = 9999
3070  X = ABS(X)
3075  P2 = PG - 2
3076  XZ=X+Z0/ZT(1,P2)
3078  XY=X+Y0/YT(1,P2)
3080  IXZ = INT(XZ * .01)
3085  IXY=  INT(XY * .01)
3090  DXZ = XZ * .01 - IXZ
3095  DXY = XY * .01 - IXY
3100  DZ = ZT(IXZ + 1, P2) - ZT(IXZ, P2)
3110  DY = YT(IXY + 1, P2) - YT(IXY, P2)
3120  SZ = ZT(IXZ, P2) + DZ * DXZ
3130  SY = YT(IXZ, P2) + DY * DXY
3140  RETURN
```

```
****************************************************************************
*                                                                          *
*   NAME:            PREPAIR                                               *
*   FUNCTION:        CLIP AND SCREEN POINT SOURCE                          *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:           NO OTHER ROUTINES                                     *
*                                                                          *
****************************************************************************

3390  D3 = D1 + (RX - C1) * S
3400  IF D1 < RY AND D2 < RY THEN DS = -2: GOTO 3440
3410  IF RX >= C1 AND RX <= C2 THEN DS = D3 - RY: GOTO 3440
3420  IF ABS(RX - C1) <= ABS(RX - C2) AND D1 >= RY THEN DS = D1 - RY: GOTO 3440
3430  DS = D2 - RY
3440  IF C1 > C2 THEN DS = -2
3450  RETURN

****************************************************************************
*                                                                          *
*   NAME:            CONC                                                  *
*   FUNCTION:        CALCULATE CONCENTRATION                               *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:           NO OTHER ROUTINES                                     *
*                                                                          *
****************************************************************************

3550  G = 0
3560  IF DS < 0 THEN GOTO 3680
3570  CP = C1
3580  DP = D1
3620  X = DP: GOSUB 3050
3630  X = H / SZ: GOSUB 2880: D = FX
3640  X = CP / SY: GOSUB 2880: C = FX
3650  G = D * C / SY / SZ + G
3670  G = G / 3.14159
3680  RETURN

****************************************************************************
*                                                                          *
*   NAME:            ROTATE                                                *
*   FUNCTION:        CALCULATE AND ROTATE COORDINATES SO WIND IS OUT OF NORTH *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:           NO OTHER ROUTINES                                     *
*                                                                          *
****************************************************************************

3690  X = DP / SY: GOSUB 2880: C = FX
3700  TH = 3.14159 / 180 * (-W): CO = COS(TH): SI = SIN(TH)
3710  C3 = C1 * CO + D1 * SI: D1 = D1 * CO - C1 * SI: C1 = C3
3720  C4 = C2 * CO + D2 * SI: D2 = D2 * CO - C2 * SI: C2 = C4
3730  C4 = RX * CO + RY * SI: RY = RY * CO - RX * SI: RX = C4
3740  C1 = C1 - RX: C2 = C2 - RX: D1 = D1 - RY: D2 = D2 - RY: RX = 0: RY = 0
3750  RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:              RECEPTOR                                             *
*   FUNCTION:          INPUT RECEPTOR FROM FILE                             *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                 *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

3904  RR = RR + 1
3905  OPEN "RECP.TXT" AS #2 LEN = 16
3906  FIELD 2, 8 AS X$, 8 AS Y$
3910  GET #2, RR
3920  XR = VAL(X$): YR = VAL(Y$)
3930  CLOSE #2
3940  RETURN

*****************************************************************************
*                                                                           *
*   NAME:              DRAW                                                 *
*   FUNCTION:          DISPLAY SOURCE                                       *
*   CALLED FROM:       DISPERSION -- DISPERSION REPORT                      *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

4510  LET KF = C1 * SM + X0
4520  LET KG = -D1 * SM + Y0
4530  IF KF < 0 THEN KF = 0
4540  IF KF > 279 THEN KF = 279
4550  IF KG < 0 THEN KG = 0
4560  IF KG > 159 THEN KG = 159
4570  LET KH = C2 * SM + X0
4580  LET KI = -D2 * SM + Y0
4590  IF KH < 0 THEN KH = 0
4600  IF KH > 279 THEN KH = 279
4610  IF KI < 0 THEN KI = 0
4620  IF KI > 159 THEN KI = 159
4650  RETURN
```

```
*************************************************************************
*                                                                       *
*   NAME:            E REPORT                                           *
*   FUNCTION:        PRINT EMISSION REPORT                              *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                               *
*   CALLS:           NO OTHER ROUTINES                                  *
*                                                                       *
*************************************************************************

5095  QX = FRE(0)
5100  'FORMAT REPORT HEADING
5150  FM$(1) = "|--------------------------------------------------------
      ---------------------------------------------------------------|"
5155  FM$(10) = "|                                                 EMISSION
      REPORT (POINT)                                                 |"
5160  FM$(2) = "|                          INPUTS
      |              OUTPUTS                                 |"
5170  FM$(11) = "|                                                      " +
      LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" + RIGHT$(YR$, 2) +
      " (" + LEFT$(STR$(HR), 2) + "00"
5180  FM$(12) = " HR.)                                                    |"
5200  FM$(3) = "|COORDINATES OF  |            INITIAL             |
      (AP-42)          |                  EMISSION RATES              |"
5210  FM$(4) = "|POINT SOURCE'S  |          PARAMETERS(M)         |-----------
      ------------|------------------------------------------------|"
5215  FM$(9) = "|ORIGIN (0,0)(M) |                                |              |"
5220  FM$(5) = "| REC|           | SIG | SIG |PLUME|TONS/|TEMP |
      |                         GM/SEC                           |"
5230  FM$(6) = "|  # |  X1    Y1  |  Z  |  Y  | HT. | HR  | (F) |            |"
5240  FM$(7) = "   CO      |    HC     |  NOX     |    SOX    |    PART   |"
5250  FM$(8) = "|----------------|--------------------------------|------------
      -----------|-----------------------------------------------|"
5260  LPRINT FM$(1)
5265  LPRINT FM$(10)
5266  LPRINT FM$(11); FM$(12)
5267  LPRINT FM$(1)
5270  LPRINT FM$(2)
5280  LPRINT FM$(1)
5290  LPRINT FM$(3)
5300  LPRINT FM$(4)
5305  LPRINT FM$(9)
5310  LPRINT FM$(8)
5320  LPRINT FM$(5); G$
5330  LPRINT FM$(6); FM$(7)
5340  LPRINT FM$(1)
5350  FOR A = 3 TO 13: FM$(A) = "": NEXT A
5351  QX = FRE(0)
5355  RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:             REPORT                                                *
*   FUNCTION:         PRINT DISPERSION REPORT                               *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:            NO OTHER ROUTINES                                     *
*                                                                           *
*****************************************************************************

5420  QX = FRE(0)
5425  FM$(9) = RIGHT$("    " + STR$(R), 4)
5430  FM$(10) = RIGHT$("     " + STR$(VAL(IN$(19))), 5)
5440  FM$(11) = "," + RIGHT$("     " + STR$(VAL(IN$(20))), 5)
5470  FM$(12) = LEFT$(IN$(8) + "     ", 5)
5480  FM$(13) = LEFT$(IN$(7) + "     ", 5)
5490  FM$(14) = LEFT$(IN$(6) + "     ", 5)
5500  FM$(15) = RIGHT$("     " + STR$(VAL(IN$(12)) * TF), 5)
5510  FM$(16) = RIGHT$("    " + STR$(TEMP), 4)
5530  FM$(17) = LEFT$(IN$(17) + "                          ", 26)
5620  FOR A = 1 TO 5: TE(A) = TE(A) + Q(A): NEXT A
5700  LPRINT"|";FM$(9);"|";FM$(10);FM$(11);"|";FM$(12);"|";FM$(13);"|";FM$(14);
      "|";FM$(15);"|";FM$(16);"|";FM$(17);"|";:FOR A=1 TO 5:
      LPRINT USING"##.##^^^^ ";Q(A);:LPRINT"|";:NEXT A:LPRINT
5706  QX = FRE(0)

5710  RETURN

*****************************************************************************
*                                                                           *
*   NAME:             PRINT                                                 *
*   FUNCTION:         PRINT CONCENTRATIONS ON THE SCREEN                    *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:            NO OTHER ROUTINES                                     *
*                                                                           *
*****************************************************************************

12020  FOR I = 1 TO 5: PC(I) = PC(I) + G * Q(I) / V: X = PC(I): GOSUB 12100:
       PRINT E$; " "; : NEXT I
12030  IF HOURN = 0 AND TWR <= 1 THEN GOSUB 5420
12040  PRINT
12100  IF X < 1E-09 THEN X = 0
12120  IF X = 0 THEN E$ = "0.00": E1 = 0: GOTO 12170
12130  E1 = LOG(X) / LOG(10)
12140  E2 = E1 - INT(E1)
12150  E3 = 10 ^ E2 + .005
12160  E$ = LEFT$(STR$(E3) + "     ", 4)
12170  E$ = E$ + "E" + RIGHT$("00" + STR$(INT(E1)), 2)
12180  RETURN
```

```
*************************************************************************
*                                                                       *
*   NAME:           D-REPORT                                            *
*   FUNCTION:       PRINT DISPERSION REPORT                             *
*   CALLED FROM:    DISPERSION -- DISPERSION SUBROUTINE                 *
*                   AND MAIN -- MAIN PROGRAM                            *
*   CALLS:          NO OTHER ROUTINES                                   *
*                                                                       *
*************************************************************************

13000  QX = FRE(0)
13001  IF HOURN >= 1 THEN 13023
13002  HOURN = HOURN + 1
13003  IF TWR > 1 THEN 13023
13004  TWR = TWR + 1
13007  LPRINT FM$(1)
13008  LPRINT "
       TOTAL |";: FOR QX = 1 TO 5:LPRINT USING "##.##^^^^ ";TE(QX);:LPRINT"|";:
       NEXT QX :LPRINT
13009  G$ = CHR$(252)
13010  FM$(2) = "|                              INPUTS                 |   |
       OUTPUT                              |"
13011  FM$(3) = "|                                            DISPERSION REPORT         |"
13016  FM$(1) = "|-------------------------------------------------------------
       ------------------------------------|"
13017  GM$(5) = "|           |   |   |   |   | NO. |  X  |  Y  |   |    CO    |
       HC     |    NOX    |    SOX    |   PART    |"
13018  GM$(1) = "|   DATE   |HR |W/S|WD |P/G|     RECEPTOR      |   |
       CONCENTRATION GM/M^3                   |"
13019  GM$(2) = "|           |   |M/S|DEG|A=1|                   |   |                |"
13020  LPRINT : LPRINT FM$(1): LPRINT FM$(3): LPRINT FM$(1): LPRINT FM$(2):

       LPRINT FM$(1): LPRINT GM$(1)
13022  LPRINT GM$(2): LPRINT FM$(1): LPRINT GM$(5)
13023  GM$(3) = "|" + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" +
       RIGHT$(YR$, 2) + "|" + LEFT$(STR$(HR)+"   ", 3) + "|"+MID$(STR$(WS)+
       "    ",2,3) + "|" + RIGHT$("    " + STR$(W), 3)
13024  GM$(4) = "|" + RIGHT$("    " + STR$(PG), 3) + "|"
13035  J2$ = RIGHT$("        " + STR$(RR) + "    ", 6)
13037  YR = INT(YR): XR = INT(XR)
13040  AD$ = RIGHT$("      " + STR$(XR), 5): AE$ = RIGHT$("      " + STR$(YR), 5)
13050  GM$(6) = J2$ + "|" + AD$ + "|" + AE$ + "|" + "    " + "|"
13051  LPRINT GM$(3);GM$(4);GM$(6);:FOR I = 1 TO 5: LPRINT USING "##.##^^^^^";
       PC(I);:LPRINT"|";: NEXT I:LPRINT
13074  GOSUB 14700
13075  FOR A = 1 TO 5: PC(A) = 0: NEXT A
13100  CLS
13105  OPEN "RECP.TXT" AS #2 LEN = 16
13106  FIELD 2, 8 AS X$, 8 AS Y$
13107  GET #2, RR + 1
13108  X = VAL(X$): Y = VAL(Y$)
13109  IF X = 0 AND Y = 0 THEN 13120
13110  IF EOF(2) THEN GOTO 13120 ELSE CLOSE #1: CLOSE #2: CLOSE #3: OPEN "I",
```

```
       #1, FACF$: OPEN "I", #3, FAC$: GOTO 122
13120  CLOSE #2: IF P1 < 5 THEN GOSUB 15000: CLOSE #1: CLOSE #2: CLOSE #3:
       TWR = 1: HOURN = 1: R = 0: OPEN "I", #1, FACF$: OPEN "I", #3, FAC$:
       RR = 0: LPRINT FM$(1): GOTO 119
13121  IF BAT$ = "I" THEN INPUT "INPUT ANOTHER HOUR "; QQ$ ELSE QQ$ = "Y"
13125  IF QQ$ <> "Y" AND QQ$ <> "N" THEN 13121
13130  IF QQ$ = "Y" THEN CLS : P1 = 0: GOSUB 15000: CLOSE #1: CLOSE #2:
       CLOSE #3: TWR = 1: HOURN = 1: R = 0: OPEN "I", #1, FACF$: OPEN "I",
       #3, FAC$: RR = 0:LPRINT FM$(1)
13132  IF QQ$ = "Y" THEN IF BAT$ = "R" THEN GOSUB 1590 ELSE GOSUB 1650
13133  IF QQ$ = "Y" THEN GOTO 120
13135  IF QQ$ = "N" THEN CLS : END
13140  RETURN

***********************************************************************
*                                                                     *
*   NAME:            CONC                                             *
*   FUNCTION:        SAVES CONCENTRATIONS FROM THIS SOURCE            *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                             *
*   CALLS:           NO OTHER ROUTINES                                *
*                                                                     *
***********************************************************************

14500  ' PRINT D$;"OPEN RECEPTOR TOTAL,L80"
14510  ' PRINT D$;"WRITE RECEPTOR,R";J1
14520  ' FOR A = 1 TO 5: PRINT PC(A): NEXT A
14540  RETURN

***********************************************************************
*                                                                     *
*   NAME:            TOTAL                                            *
*   FUNCTION:        SUMS AND SAVES TOTAL CONCENTRATIONS              *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                             *
*   CALLS:           NO OTHER ROUTINES                                *
*                                                                     *
***********************************************************************

14700  ' PRINT : PRINT D$;"OPEN RECEPTOR,D2,L170"
14710  ' PRINT D$;"READ RECEPTOR,R";J1
14720  ' FOR A = 1 TO 5: INPUT TR(A): NEXT A
14740  ' FOR A = 1 TO 5:TR(A) = TR(A) + PC(A): NEXT A
14750  ' FOR A = 1 TO 5:X = TR(A): GOSUB 12100:TR$(A) = E$: NEXT A
14760  ' PRINT D$;"OPEN RECEPTOR,L170"
14770  ' PRINT D$;"WRITE RECEPTOR,R";J1
14780  ' FOR A = 1 TO 5: PRINT TR$(A): NEXT A
14800  RETURN
```

```
******************************************************************************
*                                                                            *
*   NAME:            SETFILE                                                 *
*   FUNCTION:        SETS INPUT FILE NAMES                                   *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                    *
*   CALLS:           NO OTHER ROUTINES                                       *
*                                                                            *
******************************************************************************

15000  P1 = P1 + 1
15010  CLOSE #3
15020  ON P1 GOTO 15030, 15050, 15070, 15090, 15110
15030  FAC$ = "TEMPLANP.TXT"
15035  FACF$ = "FACFILEP.TXT"
15040  RETURN
15050  FAC$ = "TEMPLANI.TXT"
15055  FACF$ = "FACFILEI.TXT"
15060  RETURN
15070  FAC$ = "TEMPLANK.TXT"
15075  FACF$ = "FACFILET.TXT"
15080  RETURN
15090  FAC$ = "TEMPLANF.TXT"
15095  FACF$ = "FACFILEF.TXT"
15100  RETURN
15110  GOTO 13120

******************************************************************************
*                                                                            *
*   NAME:            CASE                                                    *
*   FUNCTION:        CONVERT ALL LETTERS TO UPPER CASE                       *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                    *
*   CALLS:           NO OTHER ROUTINES                                       *
*                                                                            *
******************************************************************************

15200  FOR A = 1 TO 3
15210  CONVER = ASC(MID$(M$, A, 1))
15220  IF CONVER > 96 THEN CONVER = CONVER - 32
15230  MID$(M$, A, 1) = CHR$(CONVER)
15240  NEXT A
15250  RETURN
```

```
************************************************************************
*                                                                      *
*   NAME:           PLUME RISE                                         *
*   FUNCTION:       CALCULATE PLUME RISE FOR POINT SOURCES             *
*   CALLED FROM:    MAIN -- MAIN PROGRAM                               *
*   CALLS:          NO OTHER ROUTINES                                  *
*                                                                      *
************************************************************************

15310  'PI = pi
15320  'VS = Stack flow speed (M/sec)
15330  'D  = Stack diameter (M)
15340  'g/PI = 3.12139
15350  'TS = potenial temp slop
15360  'T0 = Ambient temperature (F)
15370  'ZD = Effective source height (M)
15380  'H  = Stack or release height (M)
15390  'ZP = Plume rise (M)
15400  'XD = Downwind component of source-receptor displacement,origin at
       source (M)
15410  'D  = Source-receptor downwind distsnce (M)
15420  'XF = downwind plume displacement (M)
15430  'C  = Source-receptor crosswind distance(M)
15440  'XC = Crosswind component of source-receptor displacement,origin at
       source (M)
15450  '*****************************************************************
        *****
15460  G = 3.12139: TS = .035 * 9 / 5
15470  PI = 3.1428571#
15480  'INPUT "STACK DIAMETER           :";STACKDIA
15490  'INPUT "STACK HEIGHT             :";STHGT
15500  'INPUT "BUILDING HEIGHT          :";BLDHGT
15510  'INPUT "STACK FLOW SPEED (M/sec) :";STACFLOW
15520  'INPUT "STACK TEMPERATURE        :";STACTEMP
15530  'INPUT "AMBIENT TEMPERATURE      :" ;TEMP
15540  'INPUT "WIND SPEED (M/SEC)       :";V
15545  'INPUT "P/G CLASS (2-5)          :";PG
15550  STACKHGT = STACKHGT + BUILDHGT 'STACKHGT=STACK HEIGHT BUILDHGT =
       BUILDING HEIGHT
15560  WV = PI / 4 * STACFLOW * STACKDIA ^ 2
15570  F = G * WV * (STACTEMP - TEMP) / (TEMP + 459.67): IF F < 0 THEN F = 0
15575  IF PG > 4 THEN GOTO 15600
15580  IF F < 55 THEN XS = 14 * F ^ (5 / 8) ELSE XS = 34 * F ^ (2 / 5)
15590  ZP = 1.6 * F ^ (1 / 3) * (3.5 * XS) ^ (2 / 3) * 1 / V
15592  XF = 0
15595  GOTO 15650
15600  N = SQR(9.8 * TS * 1 / (TEMP + 459.67))
15610  XF = PI * V / N
15620  Z1 = 2.6 * (F / (WS * N ^ 2)) ^ (1 / 3)

15630  Z2 = 5! * (F ^ (1 / 4)) * (N ^ 2) ^ (-3 / 8)
15640  ZP = Z1: IF ZP > Z2 THEN ZP = Z2
15650  H = H + ZP
```

```
15660  XD = STACKDIA - XS
15670  XC = C
15700  RETURN

****************************************************************************
*                                                                          *
*   NAME:             CASE2                                                *
*   FUNCTION:         CONVERT LETTERS TO UPPER CASE                        *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                 *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

16000  FOR A=1 TO 3
16010  CONVER = ASC(MID$(M$,A,1))
16020  IF CONVER > 96 THEN CONVER = CONVER - 32
16030  MID$(m$,A,1) = CHR$(CONVER)
16040  NEXT A
16050  RETURN

****************************************************************************
*                                                                          *
*   NAME:             CASE3                                                *
*   FUNCTION:         CONVERT LETTERS TO UPPER CASE                        *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                 *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

16070  CONVER = ASC(BAT$)
16080  IF CONVER > 96 THEN CONVER = CONVER - 32
16090  BAT$ = CHR$(CONVER)
16100  RETURN

****************************************************************************
*                                                                          *
*   NAME:             END WEATHER                                          *
*   FUNCTION:         END PROGRAM WHEN END OF WEATHER FILE IS REACHED      *
*   CALLED FROM:      WEATHER SUBROUTINE                                   *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

16500  PRINT "END OF WEATHER FILE..."
16510 END
```

VARIABLE LIST

NAME	TYPE	DESCRIPTION
A	COUNTER	COUNTS OFF PT$ IN SUBSTRINGS OF 8 CHARACTERS EACH
A	COUNTER	FS(a) -- COUNTER FOR 7 DAYS OF WEEK
A$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Z (TABLE LOOKUP)
AC	REAL	SLOPE OF PLUME ENVELOPE
AN$	STRING	REPLY CHARACTER ENTERED BY USER
B$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Y (TABLE LOOKUP)
C	REAL	CROSSWIND CONCENTRATION FACTOR
C	INTEGER	ASSIGNMENT OF LEFT 2 CHARACTERS OF INPUTTED YEAR (CENTURY)
C1	REAL	CROSSWIND COORDINATE OF LINE ENDPOINT
C2	REAL	CROSSWIND COORDINATE OF LINE ENDPOINT
C3	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
C4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
chr$15	string	control character for prowriter printer (condensed mode)
CHR$27	STRING	CONTROL CHARACTER FOR PROWRITER PRINTER LINE LENGTH
CHR$81	STRING	CONTROL STRING FOR PROWRITER PRINTER LINE LENGTH
CI	REAL	CROSSWIND POINT SPACING INTERVAL FOR INTEGRATION
CP	REAL	CROSSWIND POINT DISPLACEMENT FOR INTEGRATION
CS	REAL	% COLD STARTS
CT	REAL	DOWNWIND INTERCEPT OF PLUME ENVELOPE LINE
D	INTEGER	ASSIGNMENT OF RIGHT 2 CHARACTERS OF INPUTTED YEAR(100 YRS)
D	REAL	DOWNWIND CONCENTRATION FACTOR
D%(D1)	INTEGER	D%(7) TEMP FACTOR - WEEKDAY (30% ACT = .3 * 10)
D1	REAL	DOWNWIND COORDINATE OF LINE ENDPOINT
D2	REAL	DOWNWIND COMPONENT OF LINE ENDPOINT
D3	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
D3	REAL	DOWNWIND DISTANCE USED TO COMPUTE SG

VARIABLE LIST

NAME	TYPE	DESCRIPTION
D4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
DI	REAL	DOWNWIND POINT SPACING INTERVAL FOR INTEGRATION
DP	REAL	DOWNWIND POINT LOCATION FOR INTEGRATION
DX	REAL	INTERPOLATION ARGUMENT
DY	REAL	CHANGE IN SIGMA Y BETWEEN INDICES I AND I+1
DZ	REAL	CHANGE IN SIGMA Z BETWEEN INDICES I AND I+1
F$	STRING	F$(6) SUBSCRIPTED VAR -- READS DAY OF WEEK DATA STRING
F(T,P)	real	F(5,5) EMISSION RATE (5 TEMP, 5 POL)-- TABLE LOOKUP
FC	INTEGER	DAY OF MONTH (CONVERTS DAY OF WEEK TO ONE OF 5 WEEKS IN M
FM$	STRING	FM$(A) STRINGS USED TO SET UP REPORT FORMAT (12"A" VALUES
FM$(A)	STRING	TEXT STRING FOR PRINTED OUTPUT REPORT
G	REAL	NORMALIZED CONCENTRATION
H	INTEGER	HOUR OF DAY (REDUNDENT WITH H1)
H	REAL	EFFECTIVE SOURCE HEIGHT
H%(H1)	INTEGER	H%(24) TEMP FACTOR - HOUR OF DAY (50% ACT = .[illegible] * 10)
H1	INTEGER	HOUR OF DAY
HF	INTEGER	HOURS IN YEAR CONSTANT--8760
I	COUNTER	Y/SIGMA Y RATIO * 10 -- 40 TABLE LOOKUP POINTS (4 SIGMA
I	COUNTER	DISTANCE INCREMENTAL INDEX FOR P/G TABLE LOOKUP (101 DIS
IN$	string	8 CHARACTER READ OF TEMPROAD --IN$(A) OR IN$ (K)-K &A SA
IN$	STRING	INPUT VARIED LENGTH STRINGS OF ROADFILE--C/D,%CLD,SIG,TE
IX	INTEGER	TABLE INDEX FOR INTERPOLATION
J	COUNTER	P/G CLASS INCREMENTAL INDEX 3 CLASSES (TABLE LOOKUP)
J1	COUNTER	COUNTER
K	INTEGER	GENERAL COUNTER

VARIABLE LIST

NAME	TYPE	DESCRIPTION
KF	INTEGER	Y - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KG	INTEGER	X - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KH	INTEGER	Y - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KI	INTEGER	X - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
L	REAL	LINE SOURCE LENGTH
L1	REAL	UNCLIPPED LINE SOURCE LENGTH
L2	REAL	LENGTH OF CLIPPED LINE SOURCE
M	INTEGER	ASSIGNED INTEGER VALUE OF STRING MONTH i.e."january"(1ft:
M$	STRING	USER INPUT MONTH STRING
M%(M1)	INTEGER	M%(12) TEMPORAL FACTOR - MONTH (40% ACTIVITY =.4 * 10)
M!	INTEGER	MONTH --REDUNDENT WITH "M"
ND(I)	REAL	ND(I) GAUSSIAN EXPONENTIAL VALUE (0 - 1) TABLE LOOKUP
NP	INTEGER	NUMBER OF POINTS IN LINE INTEGRATION
P	INTEGER	POLLUTANT INDEX IN F(T,P)
P2	REAL	STABILITY INDEX FOR TABLE LOOKUP
PC(I)	REAL	POLLUTANT CONCENTRATION FOR SPECIES I
PG	INTEGER	PASQUILL/GIFFORD STABILITY CLASS
PI	REAL	PI = 3.14159
PS	REAL	ROADSPEED -M/S
QF	REAL	REDUCED EMISSION RATE FOR CLIPPED LINE
RR	COUNTER	COUNTER FOR RECEPTORS
S	REAL	SLOPE OF LINE SEGMENT
SG	REAL	SIGMA Y VALUE USED TO COMPUTE POINT SPACING
SR	REAL	RATIO OF INITIAL SIGMA TO SIGMA AT RECEPTOR
T	INTEGER	TEMPERATURE INDEX IN F(P,T)

VARIABLE LIST

NAME	TYPE	DESCRIPTION
TE	real	TE(5) - TOTAL EMISSIONS - SUM OF "Q" (5 POLLUTANTS)
TEMP	REAL	USER INPUT TEMPERATURE
TH	REAL	WIND ANGLE IN RADIENS
TINDX	INTEGER	TEMPERATURE INDEX PRIOR TO TEMP INTERPOLATION
UC	REAL	CORRECTED WIND SPEED
VS	REAL	WIND COMPONENT ALONG THE ROAD
W%(W1)	INTEGER	W%(5) temp factor - monthweek (30% activity = .3 * 10)
W1	INTEGER	WEEK
WS	REAL	USER INPUT WIND SPEED
WS	REAL	WIND SPEED
X$	STRING	READ IN "x" COORD STRING FOR RECEPTOR - 8 CHAR LONG
X2	REAL	"X" COORDINATE OF OTHER LINE SOURCE ENDPOINT
XR	REAL	"X" COORDINATE OF RECEPTOR
Y$	STRING	READ IN "Y" COORD OF RECEPTOR RANDOM ACCESS 8 CHAR LON
Y0	REAL	INITIAL PLUME SIGMA Y
Y1	REAL	"Y" COORDINATE OF ROAD ENDPOINT
Y2	REAL	"Y" COORDINATE OF OTHER ENDPOINT OF ROAD
YR	REAL	"Y" COORDINATE OF RECEPTOR
YR$	STRING	USER INPUT OF "YEAR"
YT	REAL	TABULATED SIGMA Y VALUES ASSIGNED FOR P/G TABLE LOOKUP
Z0	REAL	INITIAL PLUME SIGMA Z
ZQ	INTEGER	BOOLEAN CONTROL (0 or 1) IF INPUT ERROR IN DATE, WEATHE
ZT	REAL	TABULATED SIGMA Z VALUES ASSIGNED FOR P/G TABLE LOOKU

```
****************************************************************************
*
*                         LINE SOURCE DISPERSION PROGRAM
*
****************************************************************************

10 DIM F$(6), IN$(84), F(5, 5), M%(12), W%(5), F1(15), D%(7), H%(24)
12 DIM FM$(25), PC(5), TR(5), TR$(5)
15 DIM Q(5), ND(40), TC(5, 5), ZT(100, 3), YT(100, 3), TE(5), GM$(10)
30 OPEN "I", #1, "ROADFILE.TXT"
40 OPEN "I", #3, "TEMPROAD.TXT"
50 HOURN = 0
60 INPUT "SCREENING - 'S'  OR  REFINED - 'R'"; BAT$
70 GOSUB 15070: IF BAT$ <> "S" AND BAT$ <> "R" THEN 60
80 IF BAT$ = "S" THEN 90 ELSE OPEN "I", #5, "WEATHER.DAT"
90 WIDTH "LPT1:", 132
91 LPRINT CHR$(15)
92 ' ON ERROR  GOTO 13000
93 TWR = 1: J1 = 0: VA = 0:
112 GOSUB 1470 ' ASSIGN DAY OF WEEK TO F$ ARRAY
115 GOSUB 2840 ' SET UP FOR INTERP OF GAUSS CURVE
116 GOSUB 2960 ' READ INTERPOLATION TABLE FOR SIGMAS FILE
119 GOSUB 1590 ' INPUT TIME AND WEATHER INTERACTIVELY
120 IF HOURN < 1 THEN GOSUB 5100 ' SETUP AND PRINT REPORT
122 GOSUB 3900 ' INPUT RECEPTOR LOCATION
123 GOSUB 1190 ' ZELLERS CONGRUENCE LAW
124 GOTO  1020 ' READ SOURCE
126 GOSUB 1800 ' CALCULATE TEMPORAL FACTORS
127 GOSUB 1530 ' TEMPERATURE INTERPOLATION
128 GOSUB 2600 ' CALCULATE DISPERSION
129 GOTO 124 '   CONTINUE ON TO NEXT SOURCE
130 END
140 GOSUB 3900 ' INPUT RECEPTOR LOCATION
145 GOTO 1020  ' READ SOURCE
150 END
```

```
*****************************************************************************
*
*   NAME:            FILEREAD
*   FUNCTION:        READS ROAD COORDINATES AND TRAFFIC INFORMATION
*   CALLED FROM:     MAIN -- MAIN PROGRAM
*   CALLS:           NO ROUTINES
*
*****************************************************************************
1020 R = R + 1
1030 IF EOF(1) THEN 13000
1040 RE = R - 1
1045 INPUT #3, PT$
1047 A = 0: FOR X = 1 TO LEN(PT$) STEP 8: A = A + 1: IN$(A) = MID$(PT$, X, 8)
     : NEXT X
1050 FOR K = 1 TO 25: J = K: P = 1 + INT((J - 1) / 5): T = J - (P - 1) * 5
     : F(P, T) = VAL(IN$(K)): NEXT K
1060 FOR A = 16 TO 78: INPUT #1, IN$(A): NEXT A
1065 HF = 8760: ' INPUT HF
1100 FOR K = 1 TO 7: LET D%(K) = VAL(IN$(K + 30)): NEXT K
1110 FOR K = 1 TO 5: LET W%(K) = VAL(IN$(K + 37)): NEXT K
1120 FOR K = 1 TO 12: LET M%(K) = VAL(IN$(K + 42)): NEXT K
1130 FOR K = 1 TO 24: LET H%(K) = VAL(IN$(K + 54)): NEXT K
1142 X = FRE(0)
1150 LET H = VAL(IN$(30))
1155 X = ASC(IN$(17)): IF X = 26 THEN 13000
1160 X1 = VAL(IN$(17)): Y1 = VAL(IN$(18)): X2 =
VAL(IN$(19)): Y2 = VAL(IN$(20)) 1165 PT = VAL(IN$(26)): CS
= VAL(IN$(25)): PS = VAL(IN$(24))
1170 LN = SQR((X1 - X2) ^ 2 + (Y1 - Y2) ^ 2) / 1609
1176 IN$(17) = STR$(X1): IN$(18) = STR$(Y1): IN$(19) =
STR$(X2)
     : IN$(20) = STR$(Y2): IN$(25) = STR$(CS): IN$(24) =
STR$(PS)
1177 X = FRE(0): IF VAL(IN$(24)) = 0 THEN PT = 1609:
IN$(26) = "QUEUE": PS = 5 1178 GOTO 126
```

```
*************************************************************************
*                                                                       *
*    NAME:           DAY                                                *
*                                                                       *
*    USE:            USES ZELLER'S CONGRUENCE LAW TO DETERMINE DAY OF WEEK *
*                    FROM DATE                                          *
*                                                                       *
*    CALLED FROM:    MAIN -- MAIN PROGRAM                               *
*    CALLS:          NO OTHER ROUTINES                                  *
*                                                                       *
*************************************************************************
1190 CLS
1200 C = VAL(LEFT$(YR$, 2)): D = VAL(RIGHT$(YR$, 2))
1210 GOSUB 1337
1220 M1 = INT(M): W1 = INT(FC / 7) + 1: H1 = INT(HR)
1230 M = M - 2: IF M < 1 THEN D = D - 1: M = M + 12
1240 IF D < 0 THEN D = 99 + D: C = C - 1
1250 X = (INT(2.6 * M - .2 + .00001) + FC + D + INT(D / 4 + .00001)
     + INT(C / 4 + .00001) - (2 * C))
1260 Y = X / 7
1270 Z = ABS(Y - INT(Y + .00001)) ' DECIMAL 'AINDER
1280 DT = INT(7 * Z + .00001) ' INTEGER 'AINDER
1290 DT = DT + 1
1330 RETURN
*************************************************************************
*                                                                       *
*    NAME:           MONTH                                              *
*    USE:            ASSIGNS MONTH INDEX M FOR SUBROUTINE DAY           *
*    CALLED FROM:    MAIN                                               *
*    CALLES:         NO OTHER ROUTINES                                  *
*                                                                       *
*************************************************************************
1337 M = 13
1340 IF LEFT$(M$, 3) = "JAN" THEN M = 1
1350 IF LEFT$(M$, 3) = "FEB" THEN M = 2
1360 IF LEFT$(M$, 3) = "MAR" THEN M = 3
1370 IF LEFT$(M$, 3) = "APR" THEN M = 4
1380 IF LEFT$(M$, 3) = "MAY" THEN M = 5
1390 IF LEFT$(M$, 3) = "JUN" THEN M = 6
1400 IF LEFT$(M$, 3) = "JUL" THEN M = 7
1410 IF LEFT$(M$, 3) = "AUG" THEN M = 8
1420 IF LEFT$(M$, 3) = "SEP" THEN M = 9

1430 IF LEFT$(M$, 3) = "OCT" THEN M = 10
1440 IF LEFT$(M$, 3) = "NOV" THEN M = 11
1450 IF LEFT$(M$, 3) = "DEC" THEN M = 12
1460 RETURN
```

```
***************************************************************************
*
*    NAME:           WEEK
*    USE:            SETS UP TABLE OF NAMES OF DAYS
*    CALLED FROM:    MAIN -- MAIN PROGRAM
*    CALLS:          NO OTHER ROUTINES
*
***************************************************************************
1470 FOR A = 0 TO 6: READ F$(A): NEXT A
1480 DATA "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRIDAY"
     ,"SATURDAY"
1490 RETURN
***************************************************************************
*
*    NAME:           EMISSIONS
*    USE:            INTERPOLATES ROAD EMISSIONS FOR AMBIENT TEMPERATURE
*    CALLED FROM:    MAIN -- MAIN PROGRAM
*    CALLS:          NO OTHER ROUTINES
*
***************************************************************************
1530 TINDX = INT(TEMP / 25) + 1: IF TINDX > 5 THEN TINDX = 5
1540 FOR P = 1 TO 5
1550 Q(P) = (F(TINDX, P) + (F(TINDX + 1, P) - F(TINDX, P)) *
     (TEMP / 25 - INT(TEMP / 25))) * TF
1560 NEXT P
1575 FOR I = 1 TO 5: Q(I) = Q(I) / 60: NEXT
1576 FOR A = 1 TO 5: TE(A) = TE(A) + Q(A): NEXT A
1580 RETURN
```

```
****************************************************************************
*
*    NAME:            INPUT
*    USE:             INPUT OF METEROLOGY AND DATE FROM KEYBOARD
*    CALLED FROM:     MAIN -- MAIN PROGRAM
*    CALLS:           CONDITIONALLY CALLS THE FILE INPUT VERSION OF THIS
*                     ROUTINE
*
****************************************************************************

1590 IF BAT$ = "R" THEN 2000
1592 CLS : LOCATE 3, 10: INPUT "YEAR                        = "; YR$
     : IF ZQ > 0 THEN RETURN
1593 LOCATE 5, 10: INPUT "MONTH                  = "; M$: IF ZQ > 0 THEN RETUR
1594 GOSUB 1337
1595 IF M = 13 THEN 1593: ' CHECK MONTH IS IN BOUNDS
1596 LOCATE 7, 10: INPUT "DAY OF MONTH           = ";FC: IF ZQ > 0 THEN RETURN
1600 DA = K
1610 LOCATE 9, 10: INPUT "HOUR OF DAY (1-24)     = ";HR: IF ZQ > 0 THEN RETURN
1620 LOCATE 11, 10: INPUT "TEMPERATURE (DEG F)    = ";TEMP
1622 IF TEMP > 100 THEN TEMP = 100
1624 IF TEMP < 0 THEN TEMP = 0
1625 IF ZQ > 0 THEN RETURN
1630 LOCATE 13, 10: INPUT "WIND SPEED (M/S)       = ";WS:V=WS:IF WS < 1
     THEN V = 1
1631 IF ZQ > 0 THEN RETURN
1635 LOCATE 15, 10: INPUT "WIND DIRECTION (DEG N) = ";W
1636 IF ZQ > 0 THEN RETURN
1640 LOCATE 17, 10: INPUT "P/G CLASS (2-5)        = ";PG: IF ZQ > 0 THEN RETUR
1650 CLS
1660 ZQ = 0
1664 PRINT "1.YEAR = "; YR$
1665 PRINT "2.MONTH = "; M$
1670 PRINT "3.DAY OF MONTH = "; FC
1672 IF HR > 24 THEN HR = 24
1675 PRINT "4.HOUR OF DAY = "; HR
1680 PRINT "5.TEMPERATURE = "; TEMP
1685 PRINT "6.WIND SPEED = "; WS
1690 PRINT "7.WIND DIRECTION = "; W
1695 PRINT "8.PG CLASS = "; PG
1699 X = FRE(0)
1700 INPUT "ARE THESE CORRECT (Y OR N)"; AN$
1701 IF PG < 3 THEN PG = 3
1702 IF PG > 5 THEN PG = 5
1703 IF AN$ <> "N" THEN RETURN
1704 ZQ = ZQ + 1
1705 INPUT "CHANGE WHICH VALUE"; AN
1706 CLS
1710 ON AN GOSUB 1592, 1593, 1596, 1610, 1620, 1630, 1635, 1640
1711 ZQ = 0
1720 GOTO 1650
```

```
*****************************************************************************
*                                                                           *
*    NAME:          DUTY                                                    *
*    USE:           CALCULATE THE CORRECT TEMPORAL FACTOR                   *
*    CALLED FROM:   MAIN -- MAIN PROGRAM                                    *
*    CALLS:         NO OTHER ROUTINES                                       *
*                                                                           *
*****************************************************************************

1800 TF = (M%(M1) * 10) * (W%(W1) * 10) * (D%(DT) * 10)
1810 TF = TF / 10000
1820 TF = TF * H%(H1 + 1) * 10
1830 RETURN
*****************************************************************************
*                                                                           *
*    NAME:          WEATHER                                                 *
*    USE:           FILE VERSION OF INPUT ROUTINE                           *
*    CALLED FROM:   INPUT-- SUBROUTINE                                      *
*    CALLS:         NO OTHER ROUTINES                                       *
*                                                                           *
*****************************************************************************

2000 IF EOF(5) THEN 16000
2010 INPUT #5, ENP$
2015 IF ENP$="" THEN GOTO 2000
2020 YR$ = MID$(ENP$, 1, 4): YR = VAL(YR$)
2030 M$ = MID$(ENP$, 5, 3): GOSUB 1337
2040 FC = VAL(MID$(ENP$, 8, 2))
2050 HR = VAL(MID$(ENP$, 10, 2))
2060 TEMP = VAL(MID$(ENP$, 12, 3))
2070 WS = (VAL(MID$(ENP$, 15, 2)) * .514668):V=WS:IF V<1
     THEN  V=1
2080 W = VAL(MID$(ENP$, 17, 3))
2090 PG = VAL(RIGHT$(ENP$, 1))
2110 IF PG < 2 THEN PG = 2
2120 RETURN
2130 CLOSE #5
```

```
***************************************************************************
*                                                                         *
*    NAME:            DISPERSION                                          *
*    USE:             CONTROL LOOP FOR ALL DISPERSION CALCUCATIONS        *
*    CALLED FROM:     MAIN -- MAIN PROGRAM                                *
*    CALLS:           DRAW, DUTY, ROTATE, LENGTH, DELTA, CONC, PRINT, CLIP *
*                                                                         *
***************************************************************************

2600 C1 = X1: D1 = Y1: C2 = X2: D2 = Y2
2605 GOSUB 4500 ' DRAW SOURCE
2610 GOSUB 1800 ' FACTOR ROUTINE
2620 RX = XR: RY = YR
2705 GOSUB 3700 ' ROTATE/TRANSLATE
2740 IF PG > 5 THEN PG = 5
2750 IF PG < 2 THEN PG = 2
2754 GOSUB 3340 ' DISTANCE FINDER ROUTINE
2758 UC = 1.85 * (V ^ .164) * (SIN(ATN(S)) ^ 2): IF UC > V THEN V = UC
2759 VF = ABS(V * COS(ATN(S))): Z0 = 3.57 - .53 * VF: IF Z0 < 1.5 THEN
     Z0 = 1.5
2760 Y0 = 2 * Z0
2761 GOSUB 2768
2763 GOSUB 3460 ' BREAK INTO POINTS
2764 GOSUB 3550 ' GET CONCENTRATION
2765 GOSUB 12000 ' PRINT RESULTS
2766 RETURN

***************************************************************************
*                                                                         *
*    NAME:            CLIP                                                *
*    USE:             CLIPS LINE SEGMENT FOR FASTER COMPUTATION           *
*    CALLED FROM:     DISPERSION -- DISPERSION ROUTINE                    *
*    CALLS:           LENGTH, ENVELOPE, PREPAIR, SIGMA                    *
*                                                                         *
***************************************************************************

2768 IF C1 > C2 THEN DU = C2: C2 = C1: C1 = DU: DU = D2: D2 = D1: D1 = DU
2772 GOSUB 3340 ' DISTANCE FINDER CALCULATE L1
2773 L1 = L
2774 GOSUB 3760: ' GET PLUME ENVELOPE
2778 IF Y1 > 0 AND X1 > C1 AND X1 < C2 THEN C1 = X1: D1 = Y1
2782 IF X2 > C1 AND X2 < C2 AND Y2 > 0 THEN C2 = X2: D2 = Y2
2786 GOSUB 3340 ' DISTANCE FINDER CALCULATE L2
2787 L2 = L
2788 QF = L2 / L1
2792 IF QF = 0 THEN DS = -2: GOTO 2810
2800 GOSUB 3390: X = DS
2805 GOSUB 3050: SG = SY
2810 RETURN
```

```
***********************************************************************
*                                                                     *
*    NAME:            GAUSSTABLE                                      *
*    USE:             CALCULATE THE GAUSSIAN PLUME TABLE              *
*    CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*    CALLS:           NO OTHER ROUTINES                               *
*                                                                     *
***********************************************************************

2840 FOR I = 0 TO 40
2850 ND(I) = EXP(-.005 * I * I)
2860 NEXT I
2870 RETURN

***********************************************************************
*                                                                     *
*    NAME:            NORMAL                                          *
*    USE:             CALCULATE THE NORMAL DIST CURVE                 *
*    CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*    CALLS:           NO OTHER ROUTINES                               *
*                                                                     *
***********************************************************************

2890 IF ABS(X) > 4 THEN FX = 0: GOTO 2950
2900 X = ABS(X) * 10
2910 IX = INT(X)
2920 DX = (X - IX)
2930 FA = ND(IX) - ND(IX + 1)
2940 FX = ND(IX)- FA * DX
2950 RETURN

***********************************************************************
*
*    NAME:            SIGMATABLE
*    USE:             READS IN SIGMA TABLE INFORMATION FROM FILE
*    CALLED FROM:     MAIN -- MAIN PROGRAM
*    CALLS:           NO OTHER ROUTINES
*
***********************************************************************

2960 OPEN "NEWSIGMA" FOR INPUT AS #6
2980 FOR J = 0 TO 2
2990 FOR I = 0 TO 100
3000 INPUT #6, A$, B$: ZT(I, J) = VAL(A$): YT(I, J) = VAL(B$)
3010 NEXT I
3020 NEXT J
3030 CLOSE #6
3040 RETURN
```

```
********************************************************************************
*                                                                              *
*    NAME:            SIGMA                                                    *
*    USE:             INTERPOLATION FOR SIGMAS                                 *
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                      *
*    CALLS:           NO OTHER ROUTINES                                        *
*                                                                              *
********************************************************************************

3050 IF X > 9999 THEN X = 9999
3060 X = ABS(X)
3070 P2 = PG -3:IF P2<0 THEN P2=0
3080 IX = INT(X * .01)
3090 DX = X * .01 - IX
3100 DZ = ZT(IX + 1, P2) - ZT(IX, P2)
3110 DY = YT(IX + 1, P2) - YT(IX, P2)
3120 SZ = ZT(IX, P2) + DZ * DX: IF SZ > 5 * Z0 THEN SR = Z0 / SZ
     : SZ = SZ * (1 + SR * SR * .5): GOTO 3130
3125 SZ = SQR(SZ * SZ + Z0 * Z0)
3130 SY = YT(IX, P2) + DY * DX: IF SY > 5 * Y0 THEN SR = Y0 / SY
     : SY = SY * (1 + .5 * SR * SR): GOTO 3140
3135 SY = SQR(SY * SY + Y0 * Y0)
3140 RETURN
********************************************************************************
*                                                                              *
*    NAME:            LENGTH                                                   *
*    USE:             CALCULATE LENGTH FROM COORDINATES                        *
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROURINE                      *
*    CALLS:           NO OTHER ROUTINES                                        *
*                                                                              *
********************************************************************************

3340 DX = C2 - C1: IF DX = 0 THEN DX = .000001
3350 DY = D2 - D1: IF DY = 0 THEN DY = .000001
3360 S = DY / DX
3370 L = SQR(DX * DX + DY * DY)
3380 RETURN
```

```
*************************************************************************
*                                                                       *
*    NAME:            PREPAIR                                           *
*    USE:             CLIP AND SCREEN LINE SOURCE                       *
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROURINE               *
*    CALLS:           NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

3390 D3 = D1 + (RX - C1) * S
3400 IF D1 < RY AND D2 < RY THEN DS = -2: GOTO 3440
3410 IF RX >= C1 AND RX <= C2 THEN DS = D3 - RY: GOTO 3440
3420 IF ABS(RX - C1) <= ABS(RX - C2) AND D1 >= RY THEN DS = D1 - RY: GOTO 3440
3430 DS = D2 - RY
3440 IF C1 > C2 THEN DS = -2
3450 RETURN

*************************************************************************
*                                                                       *
*    NAME:            DELTA                                             *
*    USE:             BREAK LINE INTO POINT                             *
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROURINE               *
*    CALLS:           NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

3460 NP = ABS(C2 - C1)
3470 IF NP < ABS(D2 - D1) THEN NP = ABS(D2 - D1)
3480 IF SG = 0 THEN NP = 1: GOTO 3520
3490 NP = INT(NP / SG + .5) * 2
3500 IF NP < 10 THEN NP = 10
3510 IF NP > 200 THEN NP = 200
3520 CI = (C2 - C1) / NP
3530 DI = (D2 - D1) / NP
3540 RETURN
```

```
*************************************************************************
*
*    NAME:           CONC
*    USE:            CALCULATE CONCENTRATION
*    CALLED FROM:    DISPERSION -- DISPERSION SUBROURINE
*    CALLS:          NO OTHER ROUTINES
*
*************************************************************************

3550 G = 0
3570 CP = C1 - .5 * CI - RX
3580 DP = D1 - .5 * DI - RY
3590 FOR I = 1 TO NP
3600 CP = CP + CI
3610 DP = DP + DI
3615 IF DP < 0 THEN GOTO 3660
3620 X = DP: GOSUB 3050
3630 X = H / SZ: GOSUB 2880: D = FX
3640 X = CP / SY: GOSUB 2880: C = FX
3650 G = D * C / SY / SZ + G
3660 NEXT I
3670 G = G * QF / 3.14159 / NP
3680 RETURN

*************************************************************************
*
*    NAME:           ROTATE
*    USE:            CALCULATE AND ROTATE COORDINATES SO WIND IS OUT OF NO
*    CALLED FROM:    DISPERSION -- DISPERSION SUBROURINE
*    CALLS:          NO OTHER ROUTINES
*
*************************************************************************

3700 TH = 3.14159 / 180 * (-W): CO = COS(TH): SI = SIN(TH)
3710 C3 = C1 * CO + D1 * SI: D1 = D1 * CO - C1 * SI: C1 = C3
3720 C4 = C2 * CO + D2 * SI: D2 = D2 * CO - C2 * SI: C2 = C4
3730 C4 = RX * CO + RY * SI: RY = RY * CO - RX * SI: RX = C4

3740 C1 = C1 - RX: C2 = C2 - RX: D1 = D1 - RY: D2 = D2 - RY: RX = 0: RY =
3750 RETURN
```

```
*******************************************************************************
*
*    NAME:            ENVELOPE
*    USE:             GET PLUME ENVELOPE
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE
*    CALLS:           NO OTHER ROUTINES
*
*******************************************************************************

3760 AC = .5
3770 IF PG = 4 THEN AC = .75
3780 IF PG = 5 THEN AC = 1!
3790 CT = D1 - S * C1
3800 IF AC = -S THEN X1 = C1: Y1 = D1: GOTO 3820
3810 X1 = -CT / (AC + S): Y1 = -AC * X1
3811 IF X1 > 0 OR Y1 < 0 OR X1 < C1 THEN X1 = C1: Y1 = D1: GOTO 3820
3812 X1 = X1 - 4 * Y0: Y1 = Y1 - 4 * Y0 * S
3820 IF S = AC THEN X2 = C2: Y2 = D2: GOTO 3840
3830 X2 = CT / (AC - S): Y2 = AC * X2
3831 IF X2 < 0 OR Y2 < 0 OR Y2 > C2 THEN X2 = C2: Y2 = D2: GOTO 3840
3832 X2 = X2 + 4 * Y0: Y2 = Y2 + 4 * Y0 * S
3840 RETURN

*******************************************************************************
*
*    NAME:            RECEPTOR
*    USE:             INPUT RECEPTOR FROM FILE
*    CALLED FROM:     MAIN -- MAIN PROGRAM
*    CALLS:           NO OTHER ROUTINES
*
*******************************************************************************

3900 RR = RR + 1
3905 OPEN "RECP.TXT" AS #2 LEN = 16
3906 FIELD 2,8 AS X$, 8 AS Y$
3910 GET #2 ,RR
3920 XR= VAL(X$) : YR= VAL(Y$)
3930 CLOSE #2
3940 RETURN
```

```
*****************************************************************************
*
*    NAME:             DRAW
*    USE:              DISPLAY SOURCE
*    CALLED FROM:      DISPERSION -- DISPERSION SUBROURINE
*    CALLS:            NO OTHER ROUTINES
*
*****************************************************************************

4510 LET KF = C1 * SM + X0
4520 LET KG = -D1 * SM + Y0
4530 IF KF < 0 THEN KF = 0
4540 IF KF > 279 THEN KF = 279
4550 IF KG < 0 THEN KG = 0
4560 IF KG > 159 THEN KG = 159
4570 LET KH = C2 * SM + X0
4580 LET KI = -D2 * SM + Y0
4590 IF KH < 0 THEN KH = 0
4600 IF KH > 279 THEN KH = 279
4610 IF KI < 0 THEN KI = 0
4620 IF KI > 159 THEN KI = 159
4650 RETURN
```

```
*************************************************************************
*
*    NAME:            E REPORT
*    USE:             PRINT EMISSION REPORT
*    CALLED FROM:     MAIN -- MAIN PROGRAM
*    CALLS:           NO OTHER ROUTINES
*
*************************************************************************

5100 X = FRE(0)
5110 LPRINT CHR$(27); CHR$(81)
5150 FM$(1) = "|**********************************************-
     ***************************************************************-
     ************-|"
5155 FM$(2) = "|                                                        EMISSION
     REPORT (ROADWAYS)
5160 FM$(5) = "|                          INPUTS
                               |                OUTPUTS
               |"
5165 FM$(4) = FM$(1)
5170 FM$(3) = "|                                                                 "
 + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" + RIGHT$(YR$, 2)
 + "(" + LEFT$(STR$(HR), 3) + "00"
5180 FM$(3) = FM$(3) + " HR.)
      |"
5185 FM$(6) = FM$(1)
5200 FM$(7) = "|  COORDINATES OF SOURCES (M)|     INITIAL      |
 (MOBILE 3)         |                   EMISSION RATES
5210 FM$(8) = "|    ORIGIN AT (0 , 0)       | PARAMETERS(M)    |
                  |
5215 FM$(9) = FM$(1)
5220 FM$(10) = "|ROAD|                          | SIG | SIG |PLUME|CARS/|MPH
     |%COLD|TEMP|YEAR|                           GM/SEC
5230 FM$(11) = "|  # |  X1   Y1  |  X2   Y2  |  Z  |  Y  | HT. | HR
 |    |START| (F)|    |   CO    |   HC    | NOX     |   SOX    |   PAR
5250 FM$(12) = "|***************************|***************-|******-
****************|*******************************************************
5260 FOR A = 1 TO 12: LPRINT FM$(A): NEXT A
5350 FOR A = 3 TO 13: FM$(A) = "": NEXT A
5352 X = FRE(0)
5355 RETURN
```

```
************************************************************************
*
*    NAME:            REPORT
*    USE:             PRINT DISPERSION REPORT
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE
*    CALLS:           NO OTHER ROUTINES
*
************************************************************************

5420 X = FRE(0)
5425 FM$(9) = RIGHT$("    " + STR$(R), 4)
5430 FM$(10) = RIGHT$("     " + IN$(17), 5)
5440 FM$(11) = "," + RIGHT$("     " + IN$(18), 5)
5450 FM$(12) = RIGHT$("     " + IN$(19), 5) + ","
5460 FM$(13) = RIGHT$("     " + IN$(20), 5)
5470 FM$(14) = LEFT$(STR$(Z0) + "     ", 5)
5480 FM$(15) = LEFT$(STR$(Y0) + "     ", 5)
5490 FM$(16) = LEFT$(IN$(30) + "     ", 5)
5499 IF VAL(IN$(26)) = 0 THEN 5505
5500 FM$(17) = RIGHT$("     " + STR$(INT(TF * PT + .5)), 5)
5504 GOTO 5510
5505 FM$(17) = "QUEUE"
5510 FM$(18) = RIGHT$("   " + STR$(VAL(IN$(24))), 3)
5520 FM$(19) = RIGHT$("     " + IN$(25), 5)
5530 FM$(20) = RIGHT$("    " + STR$(TEMP), 4)
5540 FM$(21) = RIGHT$("    " + YR$, 4)
5700 LPRINT "|"; FM$(9); "|"; FM$(10); FM$(11); "|"; FM$(12); FM$(13);
     "|"; FM$(14); "|"; FM$(15); "|"; FM$(16); "|"; FM$(17); "|"; FM$(18);
     "|"; FM$(19); "|"; FM$(20); "|"; FM$(21); "|";
5710 FOR A = 1 TO 5: LPRINT USING "##.##^^^^ "; Q(A); : LPRINT "|";
     : NEXT A: LPRINT
5720 X = FRE(0)
5730 RETURN

************************************************************************
*
*    NAME:            PRINT
*    USE:             PRINT CONCENTRATIONS ON THE SCREEN
*    CALLED FROM:     DISPERSION -- DISPERSION SUBROURINE
*    CALLS:           NO OTHER ROUTINES
*
************************************************************************

12020 FOR I = 1 TO 5: PC(I) = PC(I) + G * Q(I) / V: PRINT USING
      "##.##^^^^  "; PC(I); : NEXT I
12030 IF TWR <= 1 AND HOURN = 0 THEN GOSUB 5420
12040 PRINT
12050 RETURN
```

```
*************************************************************************
*
*    NAME:           D-REPORT
*    USE:            PRINT DISPERSION REPORT
*    CALLED FROM:    DISPERSION -- DISPERSION SUBROURINE
*                    AND MAIN -- MAIN PROGRAM
*    CALLS:          NO OTHER ROUTINES
*
*************************************************************************

13000 X = FRE(0)
13001 IF HOURN >= 1 THEN 13023
13002 HOURN = HOURN + 1
13003 IF TWR > 1 THEN 13023
13004 TWR = TWR + 1
13007 LPRINT FM$(1)
13008 LPRINT "
                TOTAL "; "|"; : FOR A = 1 TO 5: LPRINT USING "##.##^^^^ ";
TE(A);:LPRINT"|";: NEXT A:LPRINT
13010 FM$(2) = "|                              INPUTS                    |    |
      OUTPUT                                 |"
13011 FM$(3) = "|                                         DISPERSION REPORT
                                             |"
13016 FM$(1) = "|*****************************************************-
      ***************************************-|"
13017 GM$(5) = "|           |    |    |    |    |  NO. |   X   |   Y   |    |    CO
      |    HC    |    NOX    |    SOX    |   PART    |"
13018 GM$(1) = "|    DATE   |HR |W/S|WD |P/G|      RECEPTOR        |    |
       CONCENTRATION GM/M^3                           |"
13019 GM$(2) = "|           |    |M/S|DEG|A=1|                      |    |
                                                  |"
13020 LPRINT FM$(1): LPRINT FM$(3): LPRINT FM$(1): LPRINT FM$(2)
      : LPRINT FM$(1): LPRINT GM$(1)
13022 LPRINT GM$(2): LPRINT FM$(1): LPRINT GM$(5)
13023 GM$(3) = "|" + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2)
      + "-" + RIGHT$(YR$, 2) + "|" + LEFT$(STR$(HR)+"    ", 3) + "|" +
      MID$(STR$(WS)+"    ",2,3) + "|" + RIGHT$("    " + STR$(W), 3)
13024 GM$(4) = "|" + RIGHT$("    " + STR$(PG), 3) + "|"
13035 J2$ = RIGHT$("         " + STR$(RR) + "    ", 6)
13037 YR = INT(YR): XR = INT(XR)
13040 AD$ = RIGHT$("        " + STR$(XR), 5): AE$ = RIGHT$("        " + STR$(YR), 5
13045 LOCATE 1, 1
13050 GM$(6) = J2$ + "|" + AD$ + "|" + AE$ + "|" + "    " + "|"
13065 LPRINT GM$(3); GM$(4); GM$(6); : FOR I = 1 TO 5: LPRINT USING
      "##.##^^^^^"; PC(I); : LPRINT "|"; : NEXT I: LPRINT
13071 X = FRE(0)
13074 ' GOSUB 14700
13075 FOR A = 1 TO 5: PC(A) = 0: NEXT A
13100 CLS
13105 OPEN "RECP.TXT" AS #2 LEN = 16
13106 FIELD 2, 8 AS X$, 8 AS Y$
13107 GET #2, RR + 1
13108 X = VAL(X$): Y = VAL(Y$)
```

```
13109 IF X = 0 AND Y = 0 THEN 13120
13110 IF EOF(2) THEN GOTO 13120 ELSE CLOSE #1: CLOSE #2: CLOSE #3
      : OPEN "I", #1, "ROADFILE.TXT": OPEN "I", #3, "TEMPROAD.TXT": GOTO 140
13120 CLOSE #2
13121 IF BAT$ = "I" THEN INPUT "INPUT ANOTHER HOUR "; QQ$:   ELSE QQ$ = "Y"
13125 IF QQ$ <> "Y" AND QQ$ <> "N" THEN 13121
13130 IF QQ$ = "Y" THEN CLS : CLOSE #1: CLOSE #2: CLOSE #3
      : OPEN "I", #1, "ROADFILE.TXT": OPEN "I", #3, "TEMPROAD.TXT"
      : TWR = 1: J1 = 0: R = 0: RR = 0
13132 IF QQ$ = "Y" THEN IF BAT$ = "B" THEN GOSUB 1590 ELSE GOSUB 1650
13133 IF QQ$ = "Y" THEN GOTO 120
13135 IF QQ$ = "N" THEN CLS : END
13155 FOR A = 1 TO 5: TE(A) = 0: NEXT A
13160 CLS : GOTO 140

*************************************************************************
*
*    NAME:            OUTFILE
*    USE:             PRINT DISPERSION RESULTS TO A HOLD FILE
*    CALLED FROM:     D-REPORT -- SUBROUTINE
*    CALLS:           NO OTHER ROUTINES
*
*************************************************************************

14700 ' OPEN "A:RECEPTOR" AS #4 LEN = 170
14720 ' FOR A = 1 TO 5: INPUT#4 ,TR(A): NEXT A
14740 ' FOR A = 1 TO 5: TR(A) = TR(A) + PC(A): NEXT A
14770 ' FOR A = 1 TO 5: PRINT USING #4,"##.##^^^^"; TR(A);: NEXT A
14780 ' X =  FRE (0)
14790 ' CLOSE #4
14810 RETURN

*************************************************************************
*
*    NAME:            CASE
*    USE:             PRINT DISPERSION RESULTS TO A HOLD FILE
*    CALLED FROM:     MAIN -- MAIN PROGRAM AND INPUT
*    CALLS:           NO OTHER ROUTINES
*
*************************************************************************

15000 FOR A = 1 TO 3
15010 CONVER = ASC(MID$(M$, A, 1))
15020 IF CONVER > 96 THEN CONVER = CONVER - 32
15030 MID$(M$, A, 1) = CHR$(CONVER)
15040 NEXT A
15050 RETURN
15070 CONVER = ASC(BAT$)
15080 IF CONVER > 96 THEN CONVER = CONVER - 32
15090 BAT$ = CHR$(CONVER)
15100 RETURN
```

```
*************************************************************************
*
*    NAME:           END WEATHER
*    USE:            END PROGRAM WHEN END OF WEATHER FILE IS REACHED
*    CALLED FROM:    WEATHER -- SUBROUTINE
*    CALLS:          NO OTHER ROUTINES
*
*************************************************************************

16000 PRINT"END OF WEATHER FILE..."
16010 END
```

VARIABLE LIST

NAME	TYPE	DESCRIPTION
A	COUNTER	COUNTS OFF PT$ IN SUBSTRINGS OF 8 CHARACTERS EACH
A	COUNTER	FS(a) -- COUNTER FOR 7 DAYS OF WEEK
A$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Z (TABLE LOOKUP)
AN$	STRING	REPLY CHARACTER ENTERED BY USER
B$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Y (TABLE LOOKUP)
C	REAL	CROSSWIND CONCENTRATION FACTOR
C	INTEGER	ASSIGNMENT OF LEFT 2 CHARACTERS OF INPUTTED YEAR (CENTURY)
C4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
CD(I)	REAL	AREA UNDER GAUSS CURVE BETWEEN 0 AND I/10 (I = 0 - 40)
chr$15	string	control character for prowriter printer (condensed mode)
CHR$27	STRING	CONTROL CHARACTER FOR PROWRITER PRINTER LINE LENGTH
CHR$81	STRING	CONTROL STRING FOR PROWRITER PRINTER LINE LENGTH
CS	REAL	% COLD STARTS
D	INTEGER	ASSIGNMENT OF RIGHT 2 CHARACTERS OF INPUTTED YEAR(100 YRS)
D	REAL	DOWNWIND CONCENTRATION FACTOR
D%(D1)	INTEGER	D%(7) TEMP FACTOR - WEEKDAY (30% ACT = .3 * 10)
D4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
DX	REAL	INTERPOLATION ARGUMENT
DY	REAL	CHANGE IN SIGMA Y BETWEEN INDICES I AND I+1
DZ	REAL	CHANGE IN SIGMA Z BETWEEN INDICES I AND I+1
F$	STRING	F$(6) SUBSCRIPTED VAR -- READS DAY OF WEEK DATA STRING
F(T,P)	real	F(5,5) EMISSION RATE (5 TEMP, 5 POL)-- TABLE LOOKUP
FC	INTEGER	DAY OF MONTH (CONVERTS DAY OF WEEK TO ONE OF 5 WEEKS IN MO
FM$	STRING	FM$(A) STRINGS USED TO SET UP REPORT FORMAT (12"A" VALUES)

VARIABLE LIST

NAME	TYPE	DESCRIPTION
FM$(A)	STRING	TEXT STRING FOR PRINTED OUTPUT REPORT
G	REAL	NORMALIZED CONCENTRATION
H	INTEGER	HOUR OF DAY (REDUNDENT WITH H1)
H	REAL	EFFECTIVE SOURCE HEIGHT
H%(H1)	INTEGER	H%(24) TEMP FACTOR - HOUR OF DAY (50% ACT = .5 * 10)
H1	INTEGER	HOUR OF DAY
HF	INTEGER	HOURS IN YEAR CONSTANT--8760
I	COUNTER	Y/SIGMA Y RATIO * 10 -- 40 TABLE LOOKUP POINTS (4 SIGMA)
I	COUNTER	DISTANCE INCREMENTAL INDEX FOR P/G TABLE LOOKUP (101 DIST)
IN$	string	8 CHARACTER READ OF TEMPROAD --IN$(A) OR IN$ (K)-K &A SAME
IN$	STRING	INPUT VARIED LENGTH STRINGS OF ROADFILE--C/D,%CLD,SIG,TEMP
J	COUNTER	P/G CLASS INCREMENTAL INDEX 3 CLASSES (TABLE LOOKUP)
J1	COUNTER	COUNTER
K	INTEGER	GENERAL COUNTER
LT	REAL	TOTAL LENGTH OF ALL NORMAL LINES USED IN AREA APPROX.
M	INTEGER	ASSIGNED INTEGER VALUE OF STRING MONTH i.e."january"(1↑3)
M$	STRING	USER INPUT MONTH STRING
M%(M1)	INTEGER	M%(12) TEMPORAL FACTOR - MONTH (40% ACTIVITY =.4 * 10)
M1	INTEGER	MONTH --REDUNDENT WITH "M"
MAX	REAL	LARGEST Y-COORDINATE OF PARKING LOT CORNER
MI	REAL	SMALLEST Y COORDINATE OF PARKING LOT CORNER
NC%(J)	INTEGER	NUMBER OF ENDPOINTS FOUND FOR LINE SEGMENT J
NL	REAL	NUMBER OF LINE ELEMENTS IN INTEGRATION
P	INTEGER	POLLUTANT INDEX IN F(T,P)
P2	REAL	STABILITY INDEX FOR TABLE LOOKUP
PC(I)	REAL	POLLUTANT CONCENTRATION FOR SPECIES I

VARIABLE LIST

NAME	TYPE	DESCRIPTION
PG	INTEGER	PASQUILL/GIFFORD STABILITY CLASS
PI	REAL	PI = 3.14159
RR	COUNTER	COUNTER FOR RECEPTORS
T	INTEGER	TEMPERATURE INDEX IN F(P,T)
TE	real	TE(5) - TOTAL EMISSIONS - SUM OF "Q" (5 POLLUTANTS)
TEMP	REAL	USER INPUT TEMPERATURE
TH	REAL	WIND ANGLE IN RADIENS
TINDX	INTEGER	TEMPERATURE INDEX PRIOR TO TEMP INTERPOLATION
W%(W1)	INTEGER	W%(5) temp factor - monthweek (30% activity = .3 * 10)
W1	INTEGER	WEEK
WS	REAL	USER INPUT WIND SPEED
WS	REAL	WIND SPEED
X$	STRING	READ IN "x" COORD STRING FOR RECEPTOR - 8 CHAR LONG
XD(I)	REAL	SCREEN COORDINATES OF LINE INTEGRATION ELEMENT
XL(I)	REAL	X - COORDINATE OF NORMAL LINES IN AREA APPROXIMATION
XR	REAL	"X" COORDINATE OF RECEPTOR
XR(I)	REAL	X - COORDINATE OF PARKING LOT CORNER
Y$	STRING	READ IN "Y" COORD OF RECEPTOR RANDOM ACCESS 8 CHAR LONG
Y0	REAL	INITIAL PLUME SIGMA Y
YD(I)	REAL	SCREEN COORDINATE OF LINE INTEGRATION ELEMENT
YL(I)	REAL	Y COORDINATE OF NORMAL LINES IN AREA APPROXIMATION
YR	REAL	"Y" COORDINATE OF RECEPTOR
YR$	STRING	USER INPUT OF "YEAR"

VARIABLE LIST

NAME	TYPE	DESCRIPTION
YR(I)	REAL	Y - COORDINATE OF PARKING LOT CORNER
YT	REAL	TABULATED SIGMA Y VALUES ASSIGNED FOR P/G TABLE LOOKUP
Z0	REAL	INITIAL PLUME SIGMA Z
ZQ	INTEGER	BOOLEAN CONTROL (0 or 1) IF INPUT ERROR IN DATE, WEATHER
ZT	REAL	TABULATED SIGMA Z VALUES ASSIGNED FOR P/G TABLE LOOKUP

```
*************************************************************************
*                                                                       *
*                    AREA SOURCE DISPERSION PROGRAM                     *
*                                                                       *
*************************************************************************

10  DIM IN$(96), F(5, 5), M%(12), W%(5), F1(15), D%(7), H%(24), FM$(25)
11  DIM ND(80), TC(5, 5), ZT(100, 3), YT(100, 3), E(5), F$(6), TE(5), GM$(10)
12  DIM XL(80, 1), XQ(5), YQ(5), YL(80), NC%(80), YP(5), XP(5)
13 DIM CD(40), XD(1), YD(1), XR(10), YR(10), Q(5), PC(5)
20 OPEN "I", #1, "PARKFILE.TXT"' FILE OF SOURCES
30 OPEN "I", #3, "TEMPPARK.TXT"' CALCULATE EMISSION RATES FROM EMISSION MODEL
50 HOURN = 0
60 INPUT "SCREENING - 'S'   OR   REFINED - 'R'"; BAT$
70 GOSUB 15070: IF BAT$ <> "S" AND BAT$ <> "R" THEN 60
80 IF BAT$ = "S" THEN 90 ELSE OPEN "I", #6, "WEATHER.DAT"
90 WIDTH "LPT1:", 132 ' SET PRINTER TO 132 COLUMN PRINT MODE
91 LPRINT CHR$(15) ' SET PRINTER TO CONDENCED MODE
92 LPRINT CHR$(27); CHR$(81)
93  TWR = 1
95  J1 = 0
110  GOSUB 1470 ' ASSIGN DAYS OF WEEK TO ARRAY F$
114  GOSUB 14000 ' READ IN GAUSSIAN PLUME TABLE
115  GOSUB 2830 ' SET UP FOR INTERP OF GAUSS CURVE
116  GOSUB 2960 ' READ INTERPOLATION TABLE FOR SIGMAS
119  GOSUB 1590 ' INPUT TIME AND WEATHER
120 IF HOURN < 1 THEN GOSUB 5100: ' SETUP AND PRINT REPORT
122  GOSUB 3900 ' INPUT RECEPTOR LOCATION
123  GOSUB 1180 ' ZELLERS CONGRUENCE LAW
124  GOTO 1020  ' READ SOURCE
126  GOSUB 1800 ' CALCULATE TEMPORAL FACTOR
127  GOSUB 1500 ' TEMPERATURE INTERPOLATION
128  GOSUB 2600 ' CALCULATE DISPERSION
129  GOTO 124 ' CONTINUE TO NEXT SOURCE
130  END
```

```
*****************************************************************************
*                                                                           *
*   NAME:            FILEREAD                                               *
*   FUNCTION:        READS IN SOURCE DATA FILES                             *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                   *
*   CALLS:           NO OTHER PROGRAMS                                      *
*                                                                           *
*****************************************************************************

1020 R = R + 1
1030 IF EOF(1) OR EOF(3) THEN 13000
1040 RE = R - 1
1045 INPUT #3, PT$
1047 A = 0: FOR X = 1 TO LEN(PT$) STEP 8: A = A + 1: IN$(A) = MID$(PT$, X, 8):
     NEXT X
1050 FOR K = 1 TO 25: J = K: P = 1 + INT((J - 1) / 5): T = J - (P - 1) * 5:
F(P, T) = VAL(IN$(K)): NEXT K
1060 FOR A = 16 TO 89: INPUT #1, IN$(A): NEXT A
1065 HF = 8760: ' INPUT HOURS FULL OPERATION
1100  FOR K = 1 TO 7: LET D%(K) = 10 * VAL(IN$(K + 40)): NEXT K

1110  FOR K = 1 TO 5: LET W%(K) = 10 * VAL(IN$(K + 47)): NEXT K
1120  FOR K = 1 TO 12: LET M%(K) = 10 * VAL(IN$(K + 52)): NEXT K
1130  FOR K = 1 TO 24: LET H%(K) = 10 * VAL(IN$(K + 64)): NEXT K
1140  FOR K = 1 TO 5: LET E(K) = .0359: NEXT K
1150  LET H = VAL(IN$(38))
1174 XF = FRE(0)
1175  GOTO 126

*****************************************************************************
*                                                                           *
*   NAME:            DAY                                                    *
*   FUNCTION:        USES ZELLER'S CONGRUENCE LAW TO DETERMINE DAY OF WEEK  *
*                    FROM DATE                                              *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                   *
*   CALLS:           NO OTHER ROUTINES                                      *
*                                                                           *
*****************************************************************************

1180  CLS
1200  C = VAL(LEFT$(YR$, 2)): D = VAL(RIGHT$(YR$, 2))
1210  GOSUB 1337
1220 M1 = INT(M): W1 = INT(FC / 7) + 1: H1 = INT(HR)
1230 M = M - 2: IF M < 1 THEN D = D - 1: M = M + 12
1240  IF D < 0 THEN D = 99 + D: C = C - 1
1250 X = (INT(2.6 * M - .2 + .00001) + FC + D + INT(D / 4 + .00001) +
     INT(C/4 + .00001) - (2 * C)) 1260 Y = X / 7
1270 Z = ABS(Y - INT(Y + .00001)): REM DECIMAL REMAINDER
1280 DT = INT(7 * Z + .00001): REM INTEGER REMAINDER
1290 DT = DT + 1
1330  RETURN
```

```
**********************************************************************
*                                                                    *
*   NAME:            MONTH                                           *
*   FUNCTION:        ASSIGNS MONTH INDEX M FOR SUBROUTINE DAY        *
*   CALLED FROM:     MAIN                                            *
*   CALLS:           NO OTHER ROUTINES                               *
*                                                                    *
**********************************************************************

1337 M = 13
1338 GOSUB 15000 ' CONVERT M$ TO ALL CAPS
1340  IF LEFT$(M$, 3) = "JAN" THEN M = 1
1350  IF LEFT$(M$, 3) = "FEB" THEN M = 2
1360  IF LEFT$(M$, 3) = "MAR" THEN M = 3
1370  IF LEFT$(M$, 3) = "APR" THEN M = 4
1380  IF LEFT$(M$, 3) = "MAY" THEN M = 5
1390  IF LEFT$(M$, 3) = "JUN" THEN M = 6
1400  IF LEFT$(M$, 3) = "JUL" THEN M = 7
1410  IF LEFT$(M$, 3) = "AUG" THEN M = 8
1420  IF LEFT$(M$, 3) = "SEP" THEN M = 9
1440  IF LEFT$(M$, 3) = "NOV" THEN M = 11
1450  IF LEFT$(M$, 3) = "DEC" THEN M = 12
1460  RETURN

**********************************************************************
*                                                                    *
*   NAME:            WEEK                                            *
*   FUNCTION:        SETS UP TABLE OF NAMES OF DAYS                  *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*   CALLS:           NO OTHER ROUTINES                               *
*                                                                    *
**********************************************************************

1470  FOR A = 0 TO 6: READ F$(A): NEXT A
1480  DATA "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRIDAY","SATURDAY"
1490  RETURN

**********************************************************************
*                                                                    *
*   NAME:            EMISSIONS                                       *
*   FUNCTION:        INTERPOLATES LOT EMISSIONS FOR AMBIENT TEMPERATURE *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                            *
*   CALLS:           NO OTHER ROUTINES                               *
*                                                                    *
**********************************************************************

1530 TINDX = INT(TEMP / 25) + 1: IF TINDX > 5 THEN TINDX = 5
1540  FOR P = 1 TO 3
1550 Q(P) = (F(TINDX, P) + (F(TINDX + 1, P) - F(TINDX, P))
     * (TEMP / 25 - INT(TEMP / 25))) * TF 1560  NEXT P
1570 Q(4) = E(4) / HF * TF * 1000: Q(5) = E(5) / HF * TF * 1000
1575  FOR I = 1 TO 5: Q(I) = Q(I) / 60: NEXT
1576  FOR A = 1 TO 5: TE(A) = TE(A) + Q(A): NEXT A
1580  RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:            INPUT                                                  *
*   FUNCTION:        INPUT OF METEROLOGY AND DATE FROM KEYBOARD             *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                   *
*   CALLS:           CONDITIONALLY CALLS THE FILE INPUT VERSION OF THIS     *
*                    ROUTINE                                                *
*                                                                           *
*****************************************************************************

1590 CLS :IF BAT$ = "R" THEN 2000
1591 LOCATE 5, 10: INPUT "YEAR                          = "; YR$: IF ZQ > 0 THEN RETURN
1592 LOCATE 7, 10: INPUT "MONTH                         = "; M$
1593 GOSUB 15000: GOSUB 1337: IF ZQ > 0 THEN RETURN
1594  IF M = 13 THEN GOTO 1592: REM  CHECK MONTH IS IN BOUNDS
1595  LOCATE 9, 10: INPUT "DAY OF MONTH             = "; FC: IF ZQ > 0 THEN RETURN
1600 DA = K
1610  LOCATE 11, 10: INPUT "HOUR OF DAY (1-24)     = "; HR: IF ZQ > 0 THEN RETURN
1620  LOCATE 13, 10: INPUT "TEMPERATURE (DEG F)    = "; TEMP: IF ZQ > 0 THEN RETURN
1622  IF TEMP > 100 THEN TEMP = 100
1624  IF TEMP < 0 THEN TEMP = 0
1630  LOCATE 15, 10: INPUT "WIND SPEED (M/S)       = "; WS: V=WS: IF V < 1
      THEN WS = 1
1631  IF ZQ > 0 THEN RETURN
1635  LOCATE 17, 10: INPUT "WIND DIRECTION (DEG N) = "; W
1636  IF ZQ > 0 THEN RETURN
1640  LOCATE 19, 10: INPUT "P/G CLASS (2-5)        = "; PG: IF ZQ > 0 THEN RETURN
1650  CLS
1660 ZQ = 0
1664  PRINT "1.YEAR = "; YR$ 1665  PRINT "2.MONTH = "; M$
1670  PRINT "3.DAY OF MONTH = "; FC
1675  PRINT "4.HOUR OF DAY = "; HR
1680  PRINT "5.TEMPERATURE = "; TEMP
1685  PRINT "6.WIND SPEED ="; WS
1690  PRINT "7.DIRECTION = "; W
1695  PRINT "8.PG CLASS = "; PG
1699  XF = FRE(0)
1700  INPUT "ARE THESE CORRECT (Y OR N)"; AN$
1701  IF PG < 2 THEN PG = 2
1702  IF PG > 5 THEN PG = 5
1703  IF AN$ <> "N" THEN RETURN
1704  ZQ = ZQ + 1
1705  INPUT "CHANGE WHICH VALUE"; AN
1706  CLS
1710  ON AN GOSUB 1591, 1592, 1595, 1610, 1620, 1630, 1635, 1640
1711 ZQ = 0
1715  GOTO 1650
```

```
*************************************************************************
*                                                                       *
*   NAME:             DUTY                                              *
*   FUNCTION:         CALCULATE THE CORRECT TEMPORAL FACTOR             *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                              *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

1800 TF = M%(M1) * W%(W1) * D%(DT)
1810 TF = TF / 10000
1820 TF = TF * H%(H1 + 1)
1830  RETURN

*************************************************************************
*                                                                       *
*   NAME:             WEATHER                                           *
*   FUNCTION:         FILE VERSION OF INPUT ROUTINE                     *
*   CALLED FROM:      INPUT -- SUBROUTINE                               *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

2000 IF EOF(6) THEN 16000
2010 INPUT #6, ENP$
2020 YR$ = MID$(ENP$, 1, 4): YR = VAL(YR$)
2030 M$ = MID$(ENP$, 5, 3): GOSUB 1337
2040 FC = VAL(MID$(ENP$, 8, 2))
2050 HR = VAL(MID$(ENP$, 10, 2))
2060 TEMP = VAL(MID$(ENP$, 12, 3))
2070 WS = VAL(MID$(ENP$, 15, 2)) * .514668

2080 W = (VAL(MID$(ENP$, 17, 2)) * 10)
2090 PG = VAL(RIGHT$(ENP$, 1))
2100 IF WS < 1 THEN 2000
2110 IF PG < 3 THEN PG = 3
2120 RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:              DISPERSION                                           *
*   FUNCTION:          CONTROL LOOP FOR ALL DISPERSION CALCULATIONS         *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                 *
*   CALLS:             DRAW, ROTATE, GAUSS, LINES, CONC, PRINT              *
*                                                                           *
*****************************************************************************

2600 XQ(1) = VAL(IN$(17)): YQ(1) = VAL(IN$(18))
2610 XQ(2) = VAL(IN$(19)): YQ(2) = VAL(IN$(20))
2620 XQ(4) = VAL(IN$(21)): YQ(4) = VAL(IN$(22))
2630 XQ(3) = VAL(IN$(23)): YQ(3) = VAL(IN$(24))
2640  IF PG > 5 THEN PG = 5
2650  IF PG < 2 THEN PG = 2
2690  GOSUB 4500: REM  DRAW SOURCE
2695 RX = XR: RY = YR
2700 XF = FRE(0)

2705  GOSUB 3700: REM  ROTATE/TRANSLATE
2710  GOSUB 14270
2764  GOSUB 14045: REM GET CONCENTRATION
2765  GOSUB 12000: REM  PRINT RESULTS
2766 XF = FRE(0)
2767  RETURN

*****************************************************************************
*                                                                           *
*   NAME:              GAUSSTABLE                                           *
*   FUNCTION:          CALCULATE THE GAUSSIAN PLUME TABLE                   *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                 *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

2840  FOR I = 0 TO 40
2850 ND(I) = EXP(-.005 * I * I)

2860  NEXT I
2870  RETURN

*****************************************************************************
*                                                                           *
*   NAME:              NORMAL                                               *
*   FUNCTION:          CALCULATE THE NORMAL DIST CURVE                      *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                 *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

2890  IF ABS(X) > 4 THEN FX = 0: GOTO 2950
2900 X = ABS(X) * 10
2910 IX = INT(X)
2920 DX = (X - IX)

2930 FA = ND(IX) - ND(IX + 1)
2940 FX = ND(IX)- FA * DX
2950  RETURN
```

```
******************************************************************************
*                                                                            *
*   NAME:              SIGMATABLE                                            *
*   FUNCTION:          READS IN SIGMA TABLE INFORMATION FROM FILE            *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                  *
*   CALLS:             NO OTHER ROUTINES                                     *
*                                                                            *
******************************************************************************

2960  OPEN "SIGMA" FOR INPUT AS #2
2980  FOR J = 0 TO 3
2990  FOR I = 0 TO 100
3000  INPUT #2, A$, B$: ZT(I, J) = VAL(A$): YT(I, J) = VAL(B$)
3010  NEXT I
3020  NEXT J
3040  CLOSE #2
3045  RETURN

******************************************************************************
*                                                                            *
*   NAME:              SIGMA                                                 *
*   FUNCTION:          INTERPOLATION FOR SIGMAS                              *
*   CALLED FROM:       DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:             NO OTHER ROUTINES                                     *
*                                                                            *
******************************************************************************

3060 X = ABS(X)
3070 IF X > 9999 THEN X = 9999
3075 P2 = PG - 2:X=X+100*Z0/ZT(1,P2)
3080 IX = INT(X * .01)
3090 DX = X * .01 - IX
3100 DZ = ZT(IX + 1, P2) - ZT(IX, P2)
3110 DY = YT(IX + 1, P2) - YT(IX, P2)

3120 SZ = ZT(IX, P2) + DZ * DX
3130 SY = YT(IX, P2) + DY * DX
3140  RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:              ROTATE                                               *
*   FUNCTION:          CALCULATE AND ROTATE COORDINATES SO WIND IS OUT OF NORTH *
*   CALLED FROM:       DISPERSION -- DISPERSION SUBROUTINE                  *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

3690 X = DP / SY: GOSUB 2880: C = FX
3700 TH = 3.14159 / 180 * (-W): CO = COS(TH): SI = SIN(TH)
3710 C4 = RX * CO + RY * SI: RY = RY * CO - RX * SI: RX = C4
3720  FOR I = 1 TO 4
3730 XR(I) = XQ(I) * CO + YQ(I) * SI - RX
3740 YR(I) = YQ(I) * CO - XQ(I) * SI - RY
3742  NEXT I
3745 RX = 0: RY = 0
3750  RETURN


*****************************************************************************
*                                                                           *
*   NAME:              RECEPTOR                                             *
*   FUNCTION:          INPUT RECEPTOR FROM FILE                             *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                 *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************
3904 RR = RR + 1
3905 OPEN "R00",#2,"RECP.TXT",16
3906 FIELD 2, 8 AS X$, 8 AS Y$
3910 GET #2, RR
3920 XR = VAL(X$): YR = VAL(Y$)
3965  CLOSE #2
3970  RETURN

*****************************************************************************
*                                                                           *
*   NAME:              DRAW                                                 *
*   FUNCTION:          DISPLAY SOURCE                                       *
*   CALLED FROM:       DISPERSION -- DISPERSION SUBROUTINE                  *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

4505  FOR I = 1 TO 4
4510 XP(I) = XQ(I) * SM + X0
4520  IF XP(I) < 0 THEN XP(I) = 0
4530  IF XP(I) > 279 THEN XP(I) = 279
4540 YP(I) = Y0 - YQ(I) * SM
4550  IF YP(I) < 0 THEN YP(I) = 0
4560  IF YP(I) > 159 THEN YP(I) = 159
4570  NEXT I
4580 XP(5) = XP(1): YP(5) = YP(1)
4640  RETURN
```

```
***************************************************************************
*                                                                         *
*   NAME:               E REPORT                                          *
*   FUNCTION:           PRINT EMISSION REPORT                             *
*   CALLED FROM:        MAIN -- MAIN PROGRAM                              *
*   CALLS:              NO OTHER ROUTINES                                 *
*                                                                         *
***************************************************************************

5100 XF = FRE(0)
5150 FM$(1) = "--------------------------------------------------------------
     ----------------------------------------"
5155 FM$(13) = "|                                                     EMISSION
     REPORT (PARKING LOTS)                                            |"
5160 FM$(2) = "|
     INPUTS                                     |
     OUTPUTS                          |"
5170 FM$(11) = "|                                                      "
     + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" + RIGHT$(YR$, 2)
     + " (" + LEF T$(STR$(HR), 2) + "00"
5180 FM$(12) = " HR.)                                                         |"
5200 FM$(3) = "|COORDINATES OF SOURCES (M)                |              (MOBILE
     3)          |                    EMISSION RATES              |"
5210 FM$(4) = "|    ORIGIN AT (0 , 0)                     |
|                                          |"
5215 FM$(9) = "|-------------------------------------------------------------
     --------------|------------------------------------------------|"
5220 FM$(5) = "|LOT|                                        | SIG |PLUME|CARS/|
     MIN/|TEMP|YEAR|                         GM/SEC                 |"
5230 FM$(6) = "| # | X1    Y1 | X2    Y2 | X3    Y3 | X4    Y4 |  Z  | HT. | HR  |
     CAR | (F)|    |     CO   |     HC   |   NOX   |    SOX   |   PART  |"
5250 FM$(8) = "|------------------------------------------|------------------
     --------------|------------------------------------------------|"

5260  LPRINT FM$(1): LPRINT FM$(13): LPRINT FM$(11); FM$(12): LPRINT FM$(1):
      LPRINT FM$(2): LPRINT FM$(1): LPRINT FM$(3): LPRINT FM$(4): LPRINT
      FM$(8): LPR INT FM$(5): LPRINT FM$(6)
5340  LPRINT FM$(1)
5350  XF = FRE(0)
5355  RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:            REPORT                                                 *
*   FUNCTION:        PRINT DISPERSION REPORT                                *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                    *
*   CALLS:           NO OTHER ROUTINES                                      *
*                                                                           *
*****************************************************************************

5420 XF = FRE(0)
5425 FM$(9) = RIGHT$("    " + STR$(R), 3)
5430 FM$(10) = RIGHT$("    " + STR$(XQ(1)), 4)
5440 FM$(11) = "," + RIGHT$("    " + STR$(YQ(1)), 4)
5450 FM$(12) = RIGHT$("    " + STR$(XQ(2)), 4) + ","
5460 FM$(13) = RIGHT$("    " + STR$(YQ(2)), 4)
5461 FM$(22) = RIGHT$("    " + STR$(XQ(4)), 4) + ","
5462 FM$(23) = RIGHT$("    " + STR$(YQ(4)), 4)
5463 FM$(24) = RIGHT$("    " + STR$(XQ(3)), 4) + ","
5464 FM$(25) = RIGHT$("    " + STR$(YQ(3)), 4)
5470 FM$(14) = LEFT$(IN$(40) + "     ", 5)
5480 FM$(15) = LEFT$(" " + IN$(38) + "    ", 5)
5510 FM$(18) = RIGHT$("    " + STR$(INT(VAL(IN$(36))) * TF), 5)
5520 FM$(19) = RIGHT$("     " + STR$(VAL(IN$(35))), 5)
5530 FM$(20) = RIGHT$("    " + STR$(TEMP), 4)
5540 FM$(21) = RIGHT$("    " + STR$(VAL(IN$(33))), 4)
5700  LPRINT "|"; FM$(9); "|"; FM$(10); FM$(11); "|"; FM$(12); FM$(13);
     "|": FM$(22); FM$(23); "|"; FM$(24); FM$(25); "|"; FM$(14); "|"; FM$(15);
     "|"; FM$( 18); "|"; FM$(19); "|"; FM$(20); "|"; FM$(21); "|";
5706 XF = FRE(0)
5710 FOR I = 1 TO 5: LPRINT USING "##.##^^^^"; Q(I);:LPRINT "|";: NEXT I:LPRINT
5720  RETURN

*****************************************************************************
*                                                                           *
*   NAME:            PRINT                                                  *
*   FUNCTION:        PRINT CONCENTRATIONS ON THE SCREEN                     *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                    *
*   CALLS:           NO OTHER ROUTINES                                      *
*                                                                           *
*****************************************************************************

12020 FOR I = 1 TO 5: PC(I) = PC(I) + G * Q(I) / WS: PRINT USING "##.##^^^^^";
      PC(I); : NEXT I 12030  IF TWR <= 1 AND HOURN = 0 THEN GOSUB 5420
12040  PRINT
12100  IF X < 1E-09 THEN X = 0
12120  IF X = 0 THEN E$ = "0.00": E1 = 0: GOTO 12170
12130 E1 = LOG(X) / LOG(10)
12140 E2 = E1 - INT(E1)
12150 E3 = 10 ^ E2 + .005
12160 E$ = LEFT$(STR$(E3) + "     ", 4)
12170 E$ = E$ + "E" + RIGHT$("00" + STR$(INT(E1)), 2)
12180  RETURN
```

```
******************************************************************************
*                                                                            *
*   NAME:             D REPORT                                               *
*   FUNCTION:         PRINT DISPERSION REPORT                                *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                    *
*                     AND MAIN -- MAIN PROGRAM                               *
*   CALLS:            NO OTHER ROUTINES                                      *
*                                                                            *
******************************************************************************

13000 X = FRE(0)
13001 IF HOURN >= 1 THEN 13023
13002 HOURN = HOURN + 1
13003 IF TWR > 1 THEN 13023
13004 TWR = TWR + 1
13005  ES$ = " "
13007  LPRINT FM$(1)

13008  LPRINT "
       TOTAL :"; : FOR A = 1 TO 5: LPRINT USING"##.##^^^^";TE(A);:LPRINT":" ; :
       NEXT A
13010 FM$(2) = ":                              INPUTS                     :
      :                  OUTPUT                              :"
13011 FM$(3) = ":                                            DISPERSION
      REPORT                                                     :"
13016 FM$(1) = "-----------------------------------------------------------------
      ----------------------------------------"
13017 GM$(5) = ":             :    :    :    :    :  NO. :  X   :  Y   :    :    CO    :
      HC     :   NOX    :    SOX    :   PART    :"
13018 GM$(1) = ":    DATE    :HR :W/S:WD :P/G:     RECEPTOR        :    :
      CONCENTRATION GM/M^3                        :"
13019 GM$(2) = ":            :    :M/S:DEG:A=1:                    :
      :                                          :"
13020 LPRINT : LPRINT FM$(1): LPRINT FM$(3): LPRINT FM$(1): LPRINT FM$(2):
      LPRINT FM$(1): LPRINT GM$(1)
13022 LPRINT GM$(2): LPRINT FM$(1): LPRINT GM$(5)

13023 GM$(3) = ":" + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" +
      RIGHT$(YR$, 2) + ":" + LEFT$(STR$(HR)+"   ", 3) + ":" + RIGHT$("   " +
      STR$(WS ), 3) + ":" + RIGHT$("   " + STR$(W), 3)
13024 GM$(4) = ":" + RIGHT$("   " + STR$(PG), 3) + ":"
13035 J2$ = RIGHT$("      " + STR$(RR) + "   ", 6)
13037 YR = INT(YR): XR = INT(XR)
13040 AD$ = RIGHT$("     " + STR$(XR), 5): AE$ = RIGHT$("     " + STR$(YR), 5)
13050 GM$(6) = J2$ + ":" + AD$ + ":" + AE$ + ":" + "    " + ":"
13051  LPRINT GM$(3);GM$(4);GM$(6);:FOR I = 1 TO 5: X = PC(I): LPRINT USING
       "##.##^^^^";PC(I);:LPRINT ":";: NEXT I:LPRINT
13071 XF = FRE(0)
```

```
13074  GOSUB 14700
13075  FOR A = 1 TO 5: PC(A) = 0: NEXT A
13100 CLS

13105 OPEN "RANDOM",#2, "RECP.TXT",16
13106 FIELD 2, 8 AS X$, 8 AS Y$
13107 GET #2, RR + 1
13108 X = VAL(X$): Y = VAL(Y$)

*****************************************************************************
*                                                                           *
*   NAME:              READGAUSS                                            *
*   FUNCTION:          READ GAUSSIAN PLUME FILE INTO ARRAY                  *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                                 *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

14000  OPEN "GAUSS" FOR INPUT AS #2
14010 FOR A = 0 TO 40: INPUT #2, A$: CD(A) = VAL(A$): NEXT A
14015 CLOSE #2
14020  RETURN

*****************************************************************************
*                                                                           *
*   NAME:              CONC                                                 *
*   FUNCTION:          COMPUTES NORMALIZED CONCENTRATIONS FROM AREA SOURCE  *
*   CALLED FROM:       DISPERSION                                           *
*   CALLS:             NO OTHER ROUTINES                                    *
*                                                                           *
*****************************************************************************

14045 NL = 9
14046 G = 0
14050  FOR I = 0 TO NL
14060 X = YL(I): GOSUB 3050
14070 X = H / SZ: GOSUB 2880: D = FX
14075  IF YL(I) < 0 THEN D = 0
14080 C = 0
14090  FOR J = 0 TO 1
14100 X = XL(I, J) / SY
14110  GOSUB 14170: C = ABS(C - FX)
14120  NEXT J
14130 G = D * C / SZ + G
14140  NEXT I
14150 G = G * .79788 / LT
14160  RETURN
```

```
****************************************************************************
*                                                                          *
*   NAME:             GAUSS                                                *
*   FUNCTION:         INTERPOLATES TABLE OF AREAS UNDER GAUSS CURVE        *
*   CALLED FROM:      DISPERSION                                           *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

14170 SN = 1
14180  IF X < 0 THEN SN = -1
14190 X = ABS(X) * 10
14200  IF X > 40 THEN FX = 1: GOTO 14250
14210 IX = INT(X)
14220 DX = (X - IX)
14230 FA = CD(IX) - CD(IX + 1)
14240 FX = CD(X) - FA * DX
14250  IF SN < 1 THEN FX = 1 - FX
14260  RETURN
```

```
***************************************************************************
*                                                                         *
*   NAME:              LINES                                              *
*   FUNCTION:          APPROXIMATES AREA SOURCE USING NORMAL LINES        *
*   CALLED FROM:       DISPERSION                                         *
*   CALLS:             NO OTHER ROUTINES                                  *
*                                                                         *
***************************************************************************

14270 XR(5) = XR(1): YR(5) = YR(1)
14320 MAX = YR(1)
14330 MI = YR(1)
14340  FOR I = 0 TO 9: NC%(I) = 0: NEXT
14350  FOR I = 1 TO 4
14360  IF YR(I) > MAX THEN MAX = YR(I)
14370  IF YR(I) < MI THEN MI = YR(I)
14380  NEXT I
14390 DY = (MAX - MI) / 10
14400  FOR I = 0 TO 9
14410 YL(I) = MI + I * DY
14420  NEXT I
14430  FOR I = 1 TO 4
14440 DX = XR(I + 1) - XR(I)
14450  IF DX = 0 THEN 14480
14460 A = (YR(I + 1) - YR(I)) / DX
14470 B = YR(I) - A * XR(I)
14480  FOR J = 0 TO 9
14490  IF YL(J) > YR(I) AND YL(J) > YR(I + 1) THEN 14550
14500  IF YL(J) < YR(I) AND YL(J) < YR(I + 1) THEN 14550
14510 K = 0
14520  IF NC%(J) <> 0 THEN K = 1
14530  IF DX = 0 THEN XL(J, K) = XR(I): NC%(J) = 1: GOTO 14550
14535  IF A = 0 THEN 14550
14540 XL(J, K) = (YL(J) - B) / A: NC%(J) = 1
14550  NEXT J
14560  NEXT I

14565 LT = 0
14570  FOR I = 0 TO 9
14590  FOR J = 0 TO 1
14600 XD(J) = XL(I, J) * CO - YL(I) * SI + XR
14610 YD(J) = YL(I) * CO + XL(I, J) * SI + YR
14630 XD(J) = XD(J) * SM + XO
14640 YD(J) = -YD(J) * SM + YO
14660  NEXT J
14663 LT = LT + ABS(XL(I, 0) - XL(I, 1))
14670  NEXT I
14675  IF LT = 0 THEN LT = .00001
14680  RETURN
```

```
******************************************************************************
*                                                                            *
*   NAME:            OUTFILE                                                 *
*   FUNCTION:        PRINT DISPERSION RESULTS TO A HOLD FILE                 *
*   CALLED FROM:     D REPORT -- SUBROUTINE                                  *
*   CALLS:           NO OTHER ROUTINES                                       *
*                                                                            *
******************************************************************************

14700 'OPEN "RECEPTOR.TXT" AS #1 LEN = 170
14710 'FIELD #1,7,TR$(1),7,TR$(2),7TR$(3),7TR$(4),7,TR$(5)
14720 'FOR A = 1 TO 5: INPUT#1,J1: NEXT A
14730 'CLOSE #1
14735 'FOR A = 1 TO 5:TR(A)=VAL(TR$(A)): NEXT A
14740 'FOR A = 1 TO 5:TR(A) = TR(A) + PC(A): NEXT A
14750 'FOR A = 1 TO 5:X = TR(A): GOSUB 12100:TR$(A) = E$: NEXT A
14760 'OPEN"RECEPTOR.TXT AS #1 LEN 170
14762 'FIELD #1,7,TR$(1),7,TR$(2),7TR$(3),7TR$(4),7,TR$(5)
14765 'FOR A = 1 TO 5: TA$=TA$+TR$(A): NEXT A
14770 'PRINT #1 TA$,J1
14780 'FOR A = 1 TO 5: PRINT TR$(A): NEXT A
14800 'CLOSE #1
14810  RETURN

******************************************************************************
*                                                                            *
*   NAME:            CASE                                                    *
*   FUNCTION:        CONVERT ALL LETTERS TO UPPER CASE                       *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                    *
*   CALLS:           NO OTHER ROUTINES                                       *
*                                                                            *
******************************************************************************

15000 FOR A = 1 TO LEN(M$)
15010 CONVER = ASC(MID$(M$, A, 1))
15020 IF CONVER > 96 THEN CONVER = CONVER - 32
15030 MID$(M$, A, 1) = CHR$(CONVER)
15040 NEXT A
15050 RETURN
```

```
*************************************************************************
*                                                                       *
*   NAME:             CASE2                                             *
*   FUNCTION:         CONVERT ALL LETTERS TO UPPER CASE                 *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                              *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

15070 CONVER = ASC(BAT$)
15080 IF CONVER > 96 THEN CONVER = CONVER - 32
15090 BAT$ = CHR$(CONVER)
15100 RETURN

*************************************************************************
*                                                                       *
*   NAME:             END WEATHER                                       *
*   FUNCTION:         END PROGRAM WHEN END OF WEATHER FILE IS REACHED   *
*   CALLED FROM:      WEATHER -- SUBROUTINE                             *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

16000 PRINT"END OF WEATHER FILE..."
16010 END
```

VARIABLE LIST

NAME	TYPE	DESCRIPTION
A	COUNTER	COUNTS OFF PT$ IN SUBSTRINGS OF 8 CHARACTERS EACH
A	COUNTER	FS(a) -- COUNTER FOR 7 DAYS OF WEEK
A$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Z (TABLE LOOKUP)
AC	REAL	SLOPE OF PLUME ENVELOPE
AN$	STRING	REPLY CHARACTER ENTERED BY USER
B$	STRING	STRING VARIABLE FOR FILE READIN OF SIGMA Y (TABLE LOOKUP)
C	REAL	CROSSWIND CONCENTRATION FACTOR
C	INTEGER	ASSIGNMENT OF LEFT 2 CHARACTERS OF INPUTTED YEAR (CENTURY)
C1	REAL	CROSSWIND COORDINATE OF LINE ENDPOINT
C2	REAL	CROSSWIND COORDINATE OF LINE ENDPOINT
C3	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
C4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
chr$15	string	control character for prowriter printer (condensed mode)
CHR$27	STRING	CONTROL CHARACTER FOR PROWRITER PRINTER LINE LENGTH
CHR$81	STRING	CONTROL STRING FOR PROWRITER PRINTER LINE LENGTH
CI	REAL	CROSSWIND POINT SPACING INTERVAL FOR INTEGRATION
CP	REAL	CROSSWIND POINT DISPLACEMENT FOR INTEGRATION
CT	REAL	DOWNWIND INTERCEPT OF PLUME ENVELOPE LINE
D	INTEGER	ASSIGNMENT OF RIGHT 2 CHARACTERS OF INPUTTED YEAR(100 YRS)
D	REAL	DOWNWIND CONCENTRATION FACTOR
D%(D1)	INTEGER	D%(7) TEMP FACTOR - WEEKDAY (30% ACT = .3 * 10)
D1	REAL	DOWNWIND COORDINATE OF LINE ENDPOINT
D2	REAL	DOWNWIND COMPONENT OF LINE ENDPOINT
D3	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION

VARIABLE LIST

NAME	TYPE	DESCRIPTION
D3	REAL	DOWNWIND DISTANCE USED TO COMPUTE SG
D4	REAL	INTERMEDIATE VALUE USED IN COORDINATE ROTATION
J	COUNTER	P/G CLASS INCREMENTAL INDEX 3 CLASSES (TABLE LOOKUP)
J1	COUNTER	COUNTER
K	INTEGER	GENERAL COUNTER
KF	INTEGER	Y - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KG	INTEGER	X - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KH	INTEGER	Y - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
KI	INTEGER	X - ENDPOINT IN SCREEN COORDINATES FOR GRAPHICS
L	REAL	LINE SOURCE LENGTH
L1	REAL	UNCLIPPED LINE SOURCE LENGTH
L2	REAL	LENGTH OF CLIPPED LINE SOURCE
M	INTEGER	ASSIGNED INTEGER VALUE OF STRING MONTH i.e."january"(1ft3)
M$	STRING	USER INPUT MONTH STRING
M%(M1)	INTEGER	M%(12) TEMPORAL FACTOR - MONTH (40% ACTIVITY =.4 * 10)
M1	INTEGER	MONTH --REDUNDENT WITH "M"
ND(I)	REAL	ND(I) GAUSSIAN EXPONENTIAL VALUE (0 - 1) TABLE LOOKUP
NP	INTEGER	NUMBER OF POINTS IN LINE INTEGRATION
P2	REAL	STABILITY INDEX FOR TABLE LOOKUP
PC(I)	REAL	POLLUTANT CONCENTRATION FOR SPECIES I
PG	INTEGER	PASQUILL/GIFFORD STABILITY CLASS
PI	REAL	PI = 3.14159
Q(P)	REAL	Q(5) INTERPOLATED EMISSION RATE -- GM/S (5 POLLUTANTS)
QC(1)	REAL	CROSSWIND COMPONENT OF QUEUE ENPOINT
QD(1)	REAL	CROSSWIND COMPONENT OF QUEUE ENDPOINT

VARIABLE LIST

NAME	TYPE	DESCRIPTION
QF	REAL	REDUCED EMISSION RATE FOR CLIPPED LINE
RC(I)	REAL	CROSSWIND COMPONENT OF QUEUE ENDPOINT
RD(I)	REAL	DOWNWIND COMPONENT OF QUEUE ENDPOINT
RR	COUNTER	COUNTER FOR RECEPTORS
S	REAL	SLOPE OF LINE SEGMENT
SG	REAL	SIGMA Y VALUE USED TO COMPUTE POINT SPACING
TE	real	TE(5) - TOTAL EMISSIONS - SUM OF "Q" (5 POLLUTANTS)
TF	INTEGER	EQUIVALENT HOURS OF FULL POWER OPERATION
TH	REAL	WIND ANGLE IN RADIANS
W%(W1)	INTEGER	W%(5) temp factor - monthweek (30% activity = .3 * 10)
W1	INTEGER	WEEK
WS	REAL	USER INPUT WIND SPEED
WS	REAL	WIND SPEED
X$	STRING	READ IN "x" COORD STRING FOR RECEPTOR - 8 CHAR LONG
X2	REAL	"X" COORDINATE OF OTHER LINE SOURCE ENDPOINT
XR	REAL	"X" COORDINATE OF RECEPTOR
Y$	STRING	READ IN "Y" COORD OF RECEPTOR RANDOM ACCESS 8 CHAR LONG
Y0	REAL	INITIAL PLUME SIGMA Y
Y1	REAL	"Y" COORDINATE OF ROAD ENDPOINT
Y2	REAL	"Y" COORDINATE OF OTHER ENDPOINT OF ROAD
YR	REAL	"Y" COORDINATE OF RECEPTOR
YR$	STRING	USER INPUT OF "YEAR"
YT	REAL	TABULATED SIGMA Y VALUES ASSIGNED FOR P/G TABLE LOOKUP
Z0	REAL	INITIAL PLUME SIGMA Z

VARIABLE LIST

NAME	TYPE	DESCRIPTION
ZQ	INTEGER	BOOLEAN CONTROL (0 or 1) IF INPUT ERROR IN DATE, WEATHER
ZT	REAL	TABULATED SIGMA Z VALUES ASSIGNED FOR P/G TABLE LOOKUP

```
***************************************************************************
*                                                                         *
*                 SPECIAL LINE SOURCE DISPERSION PROGRAM                  *
*                                                                         *
***************************************************************************

10  DIM IN$(84), F(5, 5), M%(12), W%(5), F1(15), D%(7), H%(24), FM$(18)
12  DIM ND(40), TC(5, 5), ZT(100, 3), YT(100, 3), QC(1), QD(1), RC(1), RD(1)
15  DIM TE(5), F$(6), Q(5), PC(5), MF$(5), GM$(7)
20 OPEN "AIRFILE.TXT" FOR INPUT AS #1
30 OPEN "TEMPAIRR.TXT" FOR INPUT AS #2
40 OPEN "TEMPAIRQ.TXT" FOR INPUT AS #3
50 HOURN = 0
60 INPUT "SCREENING - 'S'   OR   REFINED - 'R'"; BAT$
70 GOSUB 15070: IF BAT$ <> "S" AND BAT$ <> "R" THEN 60
80 IF BAT$ = "S" THEN 85 ELSE OPEN "I", #6, "WEATHER.DAT"
85 LPRINT CHR$(27); CHR$(81)
86 WIDTH "LPT1:", 132
90 PI = 3.14159: TWR = 1: J1 = 0: VA = 0
92  ' ON ERROR GOTO 13000
95  GOSUB 1470
107  FOR A = 1 TO 5: TE(A) = 0: NEXT A
115  GOSUB 2840: ' SET UP FOR INTERP OF GAUSS CURVE
116  GOSUB 2960: ' READ INTERPOLATION TABLE FOR SIGMAS
119  GOSUB 1590: ' INPUT TIME AND WEATHER
120  IF HOURN < 1 THEN GOSUB 5150: ' SETUP AND PRINT REPORT
124  GOSUB 3900: ' INPUT RECEPTOR LOCATION
125  GOSUB 1190: ' ZELLER CONGRUENCE LAW
126  GOTO 1020: ' READ SOURCE
127  GOSUB 1800: ' TEMPORAL FACTOR CALCULATION
128  GOSUB 1530: ' TEMPERATURE INTERPOLATION
129  GOSUB 3700: ' THETA CONVERSION FROM RADIAN TO DEGREES
130  GOSUB 13200: ' CALCULATE AND PLOT POINTS OF THE RUNWAY
131  GOSUB 12000: ' CONVERT NUMBER TO CORRECT FORMAT
132  GOSUB 13400: ' CALCULATE Q(1) THROUGH Q(5)
133  GOSUB 2600: ' DISPERSION SUBPROGRAM
134  GOTO 126: ' GET NEXT RUNWAY AND QUEUE
```

```
*************************************************************************
*                                                                       *
*   NAME:             FILEREAD                                          *
*   FUNCTION:         READS IN SOURCE DATA FILES                        *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                              *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

1020 R = R + 1
1040 RE = R - 1
1045 IF EOF(1) THEN 13000
1050 INPUT #2, IN$
1052 INPUT #3, QN$
1054 K = 1
1055 FOR A = 1 TO LEN(QN$) STEP 8: IN$(K) = MID$(QN$, A, 8): K = K + 1: NEXT A
1056 FOR A = 1 TO LEN(IN$) STEP 8: IN$(K) = MID$(IN$, A, 8): K = K + 1: NEXT A
1065 HF = 8760
1070 K = 0: FOR T = 2 TO 1 STEP -1: FOR P = 1 TO 5: K = K + 1: F(T, P) =
     VAL(IN$(K)): NEXT P: NEXT T 1080 FOR T = 1 TO 2: FOR P = 1 TO 5:
     F(T, P) = F(T, P) / 60: NEXT P: NEXT T
1090 FOR A = 11 TO 78: INPUT #1, IN$(A): NEXT A
1100 FOR K = 1 TO 7: LET D%(K) = VAL(IN$(K + 30)): NEXT K
1110 FOR K = 1 TO 5: LET W%(K) = VAL(IN$(K + 37)): NEXT K
1120 FOR K = 1 TO 12: LET M%(K) = VAL(IN$(K + 42)): NEXT K
1130 FOR K = 1 TO 24: LET H%(K) = VAL(IN$(K + 54)): NEXT K
1142  LET H = VAL(IN$(28))
1151 C1 = VAL(IN$(20)): RC(0) = C1: C2 = VAL(IN$(22)): RC(1) = C2
1152 D1 = VAL(IN$(21)): RD(0) = D1: D2 = VAL(IN$(23)): RD(1) = D2
1156 QD(0) = VAL(IN$(25)): QD(1) = VAL(IN$(27))
1157 QC(0) = VAL(IN$(24)): QC(1) = VAL(IN$(26))
1160 LN = SQR(((C2 - C1) ^ 2 + (D2 - D1) ^ 2) / 1600)
1165 Y0 = VAL(IN$(29)): Z0 = VAL(IN$(30))
1173 PT = VAL(IN$(26))
1174 X = FRE(0)
1175 GOTO 127
```

```
***************************************************************************
*                                                                         *
*   NAME:              DAY                                                *
*   FUNCTION:          USES ZELLER'S CONGRUENCE LAW TO DETERMINE DAY OF WEEK  *
*                      FROM DATE                                          *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                               *
*   CALLS:             NO OTHER ROUTINES                                  *
*                                                                         *
***************************************************************************

1190  CLS
1200 C = VAL(LEFT$(YR$, 2)): D = VAL(RIGHT$(YR$, 2))
1210  GOSUB 1340
1220 M1 = INT(M): W1 = INT(FC / 7) + 1: H1 = INT(HR)
1230 M = M - 2: IF M < 1 THEN D = D - 1: M = M + 12
1240  IF D < 0 THEN D = 99 + D: C = C - 1
1250 X = (INT(2.6 * M - .2 + .00001) + FC + D + INT(D / 4 + .00001) +
     INT(C / 4 + .00001) - (2 * C)) 1260 Y = X / 7
1270 Z = ABS(Y - INT(Y + .00001)):
1280 DT = INT(7 * Z + .00001):
1290 DT = DT + 1
1330  RETURN

***************************************************************************
*                                                                         *
*   NAME:              MONTH                                              *
*   FUNCTION:          ASSIGNS MONTH INDEX M FOR SUBROUTINE DAY           *
*   CALLED FROM:       MAIN -- MAIN PROGRAM                               *
*   CALLS:             NO OTHER ROUTINES                                  *
*                                                                         *
***************************************************************************

1337 M = 13
1340  IF LEFT$(M$, 3) = "JAN" THEN M = 1
1350  IF LEFT$(M$, 3) = "FEB" THEN M = 2
1360  IF LEFT$(M$, 3) = "MAR" THEN M = 3
1370  IF LEFT$(M$, 3) = "APR" THEN M = 4
1380  IF LEFT$(M$, 3) = "MAY" THEN M = 5
1390  IF LEFT$(M$, 3) = "JUN" THEN M = 6
1400  IF LEFT$(M$, 3) = "JUL" THEN M = 7
1410  IF LEFT$(M$, 3) = "AUG" THEN M = 8
1420  IF LEFT$(M$, 3) = "SEP" THEN M = 9
1430  IF LEFT$(M$, 3) = "OCT" THEN M = 10
1440  IF LEFT$(M$, 3) = "NOV" THEN M = 11
1450  IF LEFT$(M$, 3) = "DEC" THEN M = 12
1460  RETURN
```

```
*************************************************************************
*                                                                       *
*   NAME:             WEEK                                              *
*   FUNCTION:         SETS UP TABLE OF NAMES OF DAYS                    *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                              *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

1470  FOR A = 0 TO 6: READ F$(A): NEXT A
1480  DATA  "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY","FRIDAY",
      "SATURDAY" 1490  RETURN

*************************************************************************
*                                                                       *
*   NAME:             EMISSIONS                                         *
*   FUNCTION:         PUTS EMISSIONS INTO ARRAY                         *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                              *
*   CALLS:            NO OTHER ROUTINES                                 *
*                                                                       *
*************************************************************************

1530  FOR P = 1 TO 5: Q(P) = F(1, P) * TF: NEXT P: RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:             INPUT                                                 *
*   FUNCTION:         INPUT OF METEROLOGY AND DATE FROM KEYBOARD            *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                  *
*   CALLS:            CONDITIONALLY CALLS THE FILE INPUT VERSION OF THIS    *
*                     ROUTINE                                               *
*                                                                           *
*****************************************************************************

1590 IF BAT$ = "R" THEN 2000
1592 CLS : LOCATE 3, 10: INPUT "YEAR                    = "; YR$: IF ZQ > 0 THEN
     RETURN 1593 LOCATE 5, 10: INPUT "MONTH                   = "; M$
1594 GOSUB 15000: GOSUB 1337: IF ZQ > 0 THEN RETURN
1595 IF M = 13 THEN 1593: ' CHECK MONTH IS IN BOUNDS
1596 LOCATE 7, 10: INPUT "DAY OF MONTH          = "; FC: IF ZQ > 0 THEN RETURN
1610 LOCATE 11, 10: INPUT "HOUR OF DAY (1-24)     = "; HR: IF ZQ > 0
     THEN RETURN
1630  LOCATE 13, 10: INPUT "WIND SPEED (M/S)      = "; WS: V=WS:IF V < 1
      THEN V = 1
1631  IF ZQ > 0 THEN RETURN
1635  LOCATE 15, 10: INPUT "WIND DIRECTION (DEG N) = "; W: IF ZQ > 0
      THEN RETURN 1640  LOCATE 17, 10: INPUT "P/G CLASS (2-5)        = "; PG:
      IF ZQ > 0 THEN RETURN 1641  IF PG > 5 THEN GOTO 1640
1650  CLS
1660 ZQ = 0
1665  PRINT "1.YEAR = "; YR$
1670  PRINT "2.MONTH = "; M$
1675  PRINT "3.DAY OF MONTH = "; FC
1678  IF HR > 24 THEN HR = 24
1680  PRINT "4.HOUR OF DAY = "; HR
1685  PRINT "5.WIND SPEED = ";WS
1690  PRINT "6.DIRECTION = "; W
1695  PRINT "7.PG CLASS = "; PG
1700  INPUT "ARE THESE CORRECT (Y OR N)"; AN$
1701  IF PG < 2 THEN PG = 2
1703  IF AN$ = "Y" THEN RETURN
1704  ZQ = ZQ + 1
1705  INPUT "CHANGE WHICH VALUE"; AN
1706  CLS
1710  ON AN GOSUB 1592, 1593, 1596, 1610, 1630, 1635, 1640
1711  ZQ = 0
1715  GOTO 1650

*****************************************************************************
*                                                                           *
*   NAME:             DUTY                                                  *
*   FUNCTION:         CALCULATE THE CORRECT TEMPORAL FACTOR                 *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                  *
*   CALLS:            NO OTHER ROUTINES                                     *
*                                                                           *
*****************************************************************************

1800 TF = M%(M1) * W%(W1) * D%(DT):  TF = TF * H%(H1+1): RETURN
```

```
****************************************************************************
*                                                                          *
*   NAME:            WEATHER                                               *
*   FUNCTION:        FILE VERSION OF INPUT SUBROUTINE                      *
*   CALLED FROM:     INPUT -- SUBROUTINE                                   *
*   CALLS:           NO OTHER ROUTINES                                     *
*                                                                          *
****************************************************************************

2000 IF EOF(6) THEN 16000
2010 INPUT #6, ENP$
2015 IF ENP$="" THEN GOTO 2000
2020 YR$ = MID$(ENP$, 1, 4): YR = VAL(YR$)
2030 M$ = MID$(ENP$, 5, 3): GOSUB 1337
2040 FC = VAL(MID$(ENP$, 8, 2))
2050 HR = VAL(MID$(ENP$, 10, 2))
2060 TEMP = VAL(MID$(ENP$, 12, 3))
2070 WS = VAL(MID$(ENP$, 15, 2)) * .514668
2080 W = VAL(MID$(ENP$, 17, 3))
2090 PG = VAL(RIGHT$(ENP$, 1))
2100 V=WS:IF WS<1 THEN V=1
2120 RETURN

****************************************************************************
*                                                                          *
*   NAME:            DISPERSION                                            *
*   FUNCTION:        CONTROL LOOP FOR ALL DISPERSION CALCULATIONS          *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                  *
*   CALLS:           DRAW, DUTY, ROTATE, LENGTH, DELTA, CONC, PRINT        *
*                                                                          *
****************************************************************************

2600  GOSUB 4510
2620 RX = XR: RY = YR
2705  GOSUB 3700
2740  IF PG > 5 THEN PG = 5
2750  IF PG < 2 THEN PG = 2
2760  GOSUB 2768: GOSUB 3460: GOSUB 3550: GOSUB 12000: RETURN
```

```
*****************************************************************************
*                                                                           *
*   NAME:             CLIP                                                  *
*   FUNCTION:         CLIPS LINE SEGMENT FOR FASTER COMPUTATION             *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:            LENGTH, ENVELOPE, PREPAIR, SIGMA                      *
*                                                                           *
*****************************************************************************

2768  IF C1 > C2 THEN DU = C2: C2 = C1: C1 = DU: DU = D2: D2 = D1: D1 = DU
2772  GOSUB 3340: L1 = L
2774  GOSUB 3760
2778  IF Y1 > 0 AND X1 > C1 AND X1 < C2 AND Y2 > 0 THEN C1 = X1: D1 = Y1
2782  IF Y1 > 0 AND X2 > C1 AND X2 < C2 AND Y2 > 0 THEN C2 = X2: D2 = Y2
2786  GOSUB 3340: L2 = L
2788 QF = L2 / L1
2790  IF (D1 + D2) < ABS((C1 + C2) * AC) THEN DS = -2: GOTO 2810
2792  IF QF = 0 THEN DS = -2: GOTO 2810
2800  GOSUB 3340: GOSUB 3390: X = DS: GOSUB 3060: SG = SY
2810  RETURN

*****************************************************************************
*                                                                           *
*   NAME:             GAUSSTABLE                                            *
*   FUNCTION:         CALCULATE THE GAUSSIAN PLUME TABLE                    *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                  *
*   CALLS:            NO OTHER ROUTINES                                     *
*                                                                           *
*****************************************************************************

2840  FOR I = 0 TO 40: ND(I) = EXP(-.005 * I * I): NEXT I: RETURN

*****************************************************************************
*                                                                           *
*   NAME:             NORMAL                                                *
*   FUNCTION:         CALCULATE THE NORMAL DIST CURVE                       *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                  *
*   CALLS:            NO OTHER ROUTINES                                     *
*                                                                           *
*****************************************************************************

2890  IF ABS(X) > 4 THEN FX = 0: GOTO 2950
2900 X = ABS(X) * 10
2910 IX = INT(X)
2920 DX = (X - IX)
2930 FA = ND(IX) - ND(IX + 1)
2940 FX = ND(IX)- FA * DX
2950  RETURN
```

```
**************************************************************************
*                                                                        *
*   NAME:             SIGMATABLE                                         *
*   FUNCTION:         READS IN THE SIGMA TABLE INFORMATION FROM FILE     *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                               *
*   CALLS:            NO OTHER ROUTINES                                  *
*                                                                        *
**************************************************************************

2960  OPEN "SIGMA" FOR INPUT AS #4
2980  FOR J = 0 TO 3
2990  FOR I = 0 TO 100
3000  INPUT #4, A$, B$: ZT(I, J) = VAL(A$): YT(I, J) = VAL(B$)
3010  NEXT I
3020  NEXT J
3030 CLOSE #4
3040  RETURN


**************************************************************************
*                                                                        *
*   NAME:             SIGMA                                              *
*   FUNCTION:         INTERPOLATION FOR SIGMAS                           *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                *
*   CALLS:            NO OTHER ROUTINES                                  *
*                                                                        *
**************************************************************************

3060  IF X > 9999 THEN X = 9999
3070 X = ABS(X)
3075 P2 = PG - 2
3080 IX = INT(X * .01)
3090 DX = X * .01 - IX
3100 DZ = ZT(IX + 1, P2) - ZT(IX, P2)
3110 DY = YT(IX + 1, P2) - YT(IX, P2)

3120 SZ = ZT(IX, P2) + DZ * DX: IF SZ > 5 * Z0 THEN SR = Z0 / SZ:
     SZ = SZ + SR * SR * .5: GOTO 3130 3125 SZ = SQR(SZ * SZ + Z0 * Z0)
3130 SY = YT(IX, P2) + DY * DX: IF SY > 5 * Y0 THEN SR = Y0 / SY:
     SY = SY + .5 * SR * SR: GOTO 3140 3135 SY = SQR(SY * SY + Y0 * Y0)
3140  RETURN

**************************************************************************
*                                                                        *
*   NAME:             LENGTH                                             *
*   FUNCTION:         CALCULATE LENGTH FROM COORDINATES                  *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                *
*   CALLS:            NO OTHER ROUTINES                                  *
*                                                                        *
**************************************************************************

3340 DX = C2 - C1: IF DX = 0 THEN DX = .000001
3350 DY = D2 - D1: IF DY = 0 THEN DY = .000001
3360 S = DY / DX
3370 L = SQR(DX * DX + DY * DY)
3380  RETURN
```

```
*************************************************************************
*                                                                       *
*   NAME:              PREPAIR                                          *
*   FUNCTION:          CLIP AND SCREEN SPECIAL LINE SOURCE              *
*   CALLED FROM:       DISPERSION -- DISPERSION SUBROUTINE              *
*   CALLS:             NO OTHER ROUTINES                                *
*                                                                       *
*************************************************************************

3390 D3 = D1 + (RX - C1) * S
3400  IF D1 < RY AND D2 < RY THEN DS = -2: GOTO 3440
3410  IF RX >= C1 AND RX <= C2 THEN DS = D3 - RY: GOTO 3440
3420  IF ABS(RX - C1) <= ABS(RX - C2) AND D1 >= RY THEN DS = D1 - RY: GOTO 3440
3430 DS = D2 - RY
3440  IF C1 > C2 THEN DS = -2
3450  RETURN

*************************************************************************
*                                                                       *
*   NAME:              DELTA                                            *
*   FUNCTION:          BREAK LINE INTO POINTS                           *
*   CALLED FROM:       DISPERSION -- DISPERSION SUBROUTINE              *
*   CALLS:             NO OTHER ROUTINES                                *
*                                                                       *
*************************************************************************

3460 NP = ABS(C2 - C1)
3470  IF NP < ABS(D2 - D1) THEN NP = ABS(D2 - D1)
3480  IF SG = 0 THEN NP = 1: GOTO 3520
3490 NP = INT(NP / SG + .5)*2
3500  IF NP < 1 THEN NP = 1
3510  IF NP > 200 THEN NP = 200
3520 CI = (C2 - C1) / NP
3530 DI = (D2 - D1) / NP
3540  RETURN
```

```
****************************************************************************
*                                                                          *
*   NAME:            CONC                                                  *
*   FUNCTION:        CALCULATE CONCENTRATION                               *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                   *
*   CALLS:           NO OTHER ROUTINES                                     *
*                                                                          *
****************************************************************************

3550 G = 0
3560  IF DS < 0 THEN GOTO 3680
3570 CP = C1 - .5 * CI - RX
3580 DP = D1 - .5 * DI - RY
3590  FOR I = 1 TO NP
3600 CP = CP + CI
3610 DP = DP + DI
3615  IF DP < 0 THEN GOTO 3660
3620 X = DP: GOSUB 3060

3630 X = H / SZ: GOSUB 2890: D = FX
3640 X = CP / SY: GOSUB 2890: C = FX
3650 G = D * C / SY / SZ + G
3660  NEXT I
3670 G = G * QF / PI / NP
3680  RETURN
```

```
**************************************************************************
*                                                                        *
*   NAME:            ROTATE                                              *
*   FUNCTION:        CALCULATE AND ROTATE COORDINATES SO WIND IS OUT OF NORTH *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                 *
*   CALLS:           NO OTHER ROUTINES                                   *
*                                                                        *
**************************************************************************

3700 TH = PI / 180 * (-W): CO = COS(TH): SI = SIN(TH)
3701 RX = XR: RY = YR
3710 C3 = C1 * CO + D1 * SI: D1 = D1 * CO - C1 * SI: C1 = C3
3720 C4 = C2 * CO + D2 * SI: D2 = D2 * CO - C2 * SI: C2 = C4
3730 C4 = RX * CO + RY * SI: RY = RY * CO - RX * SI: RX = C4
3740 C1 = C1 - RX: C2 = C2 - RX: D1 = D1 - RY: D2 = D2 - RY: RX = 0: RY = 0
3750  RETURN

**************************************************************************
*                                                                        *
*   NAME:            ENVELOPE                                            *
*   FUNCTION:        GET PLUME ENVELOPE                                  *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                 *
*   CALLS:           NO OTHER ROUTINES                                   *
*                                                                        *
**************************************************************************

3760 AC = 1.5: IF PG = 3 THEN AC = 2
3770  IF PG = 4 THEN AC = 3
3780  IF PG = 5 THEN AC = 4
3790 CT = D1 - S * C1
3800  IF AC = -S THEN X1 = C1: Y1 = D1: GOTO 3820
3810 X1 = -CT / (AC + S): Y1 = -AC * X1
3815  IF X1 > 0 OR Y1 < 0 THEN X1 = C1: Y1 = D1
3820  IF S = AC THEN X2 = C2: Y2 = D2: GOTO 3840
3830 X2 = CT / (AC - S): Y2 = AC * X2
3835  IF X2 < 0 OR Y2 < 0 THEN X2 = C2: Y2 = D2
3840  RETURN
```

```
*******************************************************************************
*                                                                             *
*   NAME:            E-REPORT                                                 *
*   FUNCTION:        PRINT EMISSION REPORT                                    *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                     *
*   CALLS:           NO OTHER ROUTINES                                        *
*                                                                             *
*******************************************************************************

5150 FM$(1) = "|-----------------------------------------------------------------
     ---------------------------------------------------------------|"
5155 FM$(10) = "|                                                           EMISS
     ION REPORT                                                     |"
5160 FM$(2) = "|                      INPUTS
       |           OUTPUTS                                         |"
5170 FM$(11) = "|                                                              "
     + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" + RIGHT$(YR$, 2)
     + " (" + LEFT$(STR$(HR), 2) + "00"
5180 FM$(12) = " HR.)                                                          |"
5200 FM$(3) = "|  COORDINATES OF SOURCES (M)|     INITIAL     |          (AP-42)
       |                    EMISSION RATES                         |"

5210 FM$(4) = "|    ORIGIN AT (0 , 0)       | PARAMETERS(M)   |
       |                                                           |"
5215 FM$(9) = "|-----------------------------------------------------------------
     ---------|-----------------------------------------------------|"
5220 FM$(5) = "| REC|                         | SIG |PLUME| SIG |ACFT/|   AIRCRAF
     T TYPE   |                   GM/SEC                            |"
5230 FM$(6) = "|  # |  X1    Y1  |  X2   Y2   |  Y  | HT. |  Z  | HR  |
        |"
5240 FM$(7) = "   CO      |    HC     |  NOX     |    SOX    |    PART    |"
5250 FM$(8) = "|--------------------------|-----------------|-----------------
     ---------|-----------------------------------------------------|"
5260  LPRINT FM$(1): LPRINT FM$(10): LPRINT FM$(11); FM$(12): LPRINT FM$(9):
      LPRINT FM$(2): LPRINT FM$(9) 5290  LPRINT FM$(3): LPRINT FM$(4):
      LPRINT FM$(8): LPRINT FM$(5): LPRINT FM$(6); FM$(7): LPRINT FM$(1)
5350  FOR A = 3 TO 12: FM$(A) = "": NEXT A
5351 X = FRE(0)
5355  RETURN
```

```
***************************************************************************
*                                                                         *
*   NAME:            REPORT                                               *
*   FUNCTION:        PRINT DISPERSION REPORT                              *
*   CALLED FROM:     DISPERSION -- DISPERSION SUBROUTINE                  *
*   CALLS:           NO OTHER ROUTINES                                    *
*                                                                         *
***************************************************************************

5420 X = FRE(0)
5425 FM$(9) = RIGHT$("    " + STR$(R), 4)
5430 FM$(10) = RIGHT$("     " + STR$(VAL(IN$(20))), 5)
5440 FM$(11) = "," + RIGHT$("     " + STR$(VAL(IN$(21))), 5)
5450 FM$(12) = RIGHT$("     " + STR$(VAL(IN$(22))), 5) + ","
5460 FM$(13) = RIGHT$("     " + STR$(VAL(IN$(23))), 5)
5470 FM$(14) = LEFT$(IN$(29) + "     ", 5)
5480 FM$(15) = LEFT$(IN$(28) + "     ", 5)
5490 FM$(16) = LEFT$(IN$(30) + "      ", 5)
5491 TI = TI + 1: IF TI / 2 = INT(TI / 2) THEN GOTO 5500
5495 IN$(12) = STR$(VAL(IN$(12)) * TF)
5500 FM$(17) = RIGHT$("     " + IN$(12), 5)
5510 FM$(18) = RIGHT$("                                  " + IN$(11), 19)
5620  FOR A = 1 TO 5: TE(A) = TE(A) + Q(A): NEXT A
5700  LPRINT "|"; FM$(9); "|"; FM$(10); FM$(11); "|"; FM$(12); FM$(13); "|";
      FM$(14); "|"; FM$(15); "|"; FM$(16); "|"; FM$(17); "|"; FM$(18); "|";
5710  FOR A = 1 TO 5: LPRINT USING "##.##^^^^ "; Q(A); : LPRINT "|"; :
      NEXT A: LPRINT
5720 X = FRE(0)
5730  RETURN
```

```
**************************************************************************
*                                                                        *
*   NAME:             PRINT                                              *
*   FUNCTION:         PRINT CONCENTRATIONS ON THE SCREEN                 *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                *
*   CALLS:            NO OTHER ROUTINES                                  *
*                                                                        *
**************************************************************************

12000  FOR I = 1 TO 5: PC(I) = PC(I) + G * Q(I) / V: X = PC(I):
       NEXT I
12010  PRINT
12030  IF TWR <= 1 THEN GOSUB 5420
12040  RETURN

**************************************************************************
*                                                                        *
*   NAME:             D-REPORT                                           *
*   FUNCTION:         PRINT DISPERSION REPORT                            *
*   CALLED FROM:      DISPERSION -- DISPERSION SUBROUTINE                *
*                     AND MAIN -- MAIN PROGRAM                           *
*   CALLS:            NO OTHER ROUTINES                                  *
*                                                                        *
**************************************************************************

13000  X = FRE(0)
13002 HOURN = HOURN + 1
13003  IF TWR > 1 THEN 13023
13004 TWR = TWR + 1
13005 ES$ = " "
13007  LPRINT FM$(1)
13008  LPRINT "
       TOTAL  !";
13009 FOR A = 1 TO 5: LPRINT USING "##.##^^^^ "; TE(A); :
       LPRINT "!"; : NEXT A
13010 FM$(2) = "!                              INPUTS                  !    !
      OUTPUT                                  !"
13011 FM$(3) = "!                                          DISPERSION REPORT
                                        !"
13016 FM$(1) = "!-----------------------------------------------------------
-----------------------------------------!"
13017 GM$(5) = "!          !   !   !   !   !  NO. !  X  !  Y  !   !    CO    !
      HC    !    NOX    !    SOX    !   PART    !"
13018 GM$(1) = "!   DATE   !HR !W/S!WD !P/G!     RECEPTOR     !   !        CONCE
      NTRATION GM/M^3                      !"
13019 GM$(2) = "!          !   !M/S!DEG!A=1!                  !   !
                                        !"
13020 LPRINT FM$(1): LPRINT FM$(3): LPRINT FM$(1): LPRINT FM$(2):
      LPRINT FM$(1): LPRINT GM$(1)
13022  LPRINT GM$(2): LPRINT FM$(1): LPRINT GM$(5)
13023 GM$(3) = "!" + LEFT$(M$, 3) + "-" + RIGHT$("  " + STR$(FC), 2) + "-" +
      RIGHT$(YR$, 2) + "!" + LEFT$(STR$(HR)+"   ", 3) + "!" +
      MID$(STR$(WS)+"   ", 2,3) + "!" + RIGHT$("   " + STR$(W), 3)
```

```
13024 GM$(4) = "¦" + RIGHT$("   " + STR$(PG), 3) + "¦"
13035 J2$ = RIGHT$("      " + STR$(RR) + "   ", 6)
13037 YR = INT(YR): XR = INT(XR)
13040 AD$ = RIGHT$("     " + STR$(XR), 5): AE$ = RIGHT$("     " + STR$(YR), 5)
13050 GM$(6) = J2$ + "¦" + AD$ + "¦" + AE$ + "¦" + "   " + "¦"
13060  LPRINT GM$(3); GM$(4); GM$(6);
13070  FOR X = 1 TO 5: LPRINT USING "##.##^^^^ "; PC(X); : LPRINT "¦";
       : NEXT X: LPRINT 13074  GOSUB 14700
13075  FOR A = 1 TO 5: PC(A) = 0: NEXT A
13100 CLS
13105 OPEN "RECP.TXT" AS #4 LEN = 16
13106 FIELD 4, 8 AS X$, 8 AS Y$
13107 GET #4, RR + 1
13108 X = VAL(X$): Y = VAL(Y$)
13109 IF X = 0 AND Y = 0 THEN 13120
13110 IF EOF(4) THEN GOTO 13120 ELSE CLOSE #1: CLOSE #2: CLOSE #3: CLOSE #4:
      OPEN "I", #1, "AIRFILE.TXT": OPEN "I", #2, "TEMPAIRR.TXT":
      OPEN "I", #3, "TEMP AIRQ.TXT": GOTO 124
13120 CLOSE #4
13122 IF BAT$ = "S" THEN INPUT "INPUT ANOTHER HOUR "; QQ$:   ELSE QQ$ = "Y"
13125  IF QQ$ <> "Y" AND QQ$ <> "N" THEN 13122
13130  IF QQ$ = "Y" THEN CLS:R = 0:RR = 0:CLOSE #1:CLOSE #2:CLOSE #3:CLOSE #4:
       OPEN "I", #1,"AIRFILE.TXT":OPEN "I", #2,"TEMPAIRR.TXT":OPEN "I",
       #3,"TEMPAIRQ .TXT"
13132 IF QQ$ = "Y" THEN IF BAT$ = "R" THEN GOSUB 1590 ELSE GOSUB 1650
13133 IF QQ$ = "Y" THEN GOTO 124
13135 IF QQ$ = "N" THEN CLS : END
```

```
*****************************************************************************
*                                                                           *
*   NAME:            RUNWAY                                                 *
*   FUNCTION:        CALCULATE DISPERSION FROM ACCELERATING AIRPLANE        *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                   *
*   CALLS:           NO OTHER ROUTINES                                      *
*                                                                           *
*****************************************************************************

13200 G = 0: NP =359: QI = 0
13210  IF D1 > D2 THEN DU = D2: D2 = D1: D1 = DU: DU = C2: C2 = C1: C1 = DU:
       QI = 1
13220 CI = (C2 - C1) / NP / NP: DI = (D2 - D1) / NP / NP
13240  FOR I = 0 TO NP - 1
13250 CP = C1 + (I * I - I +.5) * CI
13260 DP = D1 + (I * I - I +.5) * DI
13265 CU = CP * CO - DP * SI + XR
13266 DU = DP * CO + CP * SI + YR
13267 DU = YO - DU * SM
13268 CU = CU * SM + XO
13270  IF DP < 0 THEN 13305
13275 X = DP: GOSUB 3060
13280 X = H / SZ: GOSUB 2890: D = FX
13290 X = CP / SY: GOSUB 2890: C = FX
13300 G = D * C / SY / SZ + G
13305  IF I = 0 THEN KF = CU: KG = DU
13310  NEXT I
13320 G = G / PI / (NP - 1)
13330  RETURN

*****************************************************************************
*                                                                           *
*   NAME:            QUEUESET                                               *
*   FUNCTION:        SETS UP FOR QUEUE CALCULATION                          *
*   CALLED FROM:     MAIN -- MAIN PROGRAM                                   *
*   CALLS:           NO OTHER ROUTINES                                      *
*                                                                           *
*****************************************************************************

13400 C2 = QC(QI): D2 = QD(QI): C1 = RC(QI): D1 = RD(QI)
13410  FOR P = 1 TO 5: Q(P) = F(2, P) * TF: NEXT P
13420 IN$(20) = STR$(C2): IN$(21) = STR$(D2): IN$(22) = STR$(C1): IN$(23) =
      STR$(D1)
13430  RETURN
```

```
****************************************************************************
*                                                                          *
*   NAME:             OUTFILE                                              *
*   FUNCTION:         PRINT DISPERSION RESULTS TO A HOLD FILE              *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                 *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

14700 '  OPEN "RECEPTOR" AS #4 LEN=170
14705 ' IF J1<2 THEN 14740
14720 '  FOR A = 1 TO 5: INPUT#4, TR(A): NEXT A
14740 '  FOR A = 1 TO 5:TR(A) = PC(A) + TR(A): NEXT A
14780 '  FOR A = 1 TO 5: PRINT USING #4,"##.##^^^^";TR(A);: NEXT A
14800 '  CLOSE #4
14805 '  FOR A = 1 TO 5:TR$(A) = "": NEXT A
14806 ' X =  FRE (0)
14810  RETURN

****************************************************************************
*                                                                          *
*   NAME:             CASE                                                 *
*   FUNCTION:         CONVERT LETTERS TO UPPER CASE                        *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                 *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

15000 FOR A = 1 TO 3
15010 CONVER = ASC(MID$(M$, A, 1))
15020 IF CONVER > 96 THEN CONVER = CONVER - 32
15030 MID$(M$, A, 1) = CHR$(CONVER)
15040 NEXT A
15050 RETURN

****************************************************************************
*                                                                          *
*   NAME:             CASE2                                                *
*   FUNCTION:         CONVERT LETTERS TO UPPER CASE                        *
*   CALLED FROM:      MAIN -- MAIN PROGRAM                                 *
*   CALLS:            NO OTHER ROUTINES                                    *
*                                                                          *
****************************************************************************

15070 CONVER = ASC(BAT$)
15080 IF CONVER > 96 THEN CONVER = CONVER - 32
15090 BAT$ = CHR$(CONVER)
15100 RETURN
```

```
************************************************************************
*                                                                      *
*   NAME:           CASE3                                              *
*   FUNCTION:       CONVERT LETTERS TO UPPER CASE                      *
*   CALLED FROM:    MAIN -- MAIN PROGRAM                               *
*   CALLS:          NO OTHER ROUTINES                                  *
*                                                                      *
************************************************************************

15110 CONVER = ASC(AN$)
15120 IF CONVER > 96 THEN CONVER = CONVER - 32
15130 AN$ = CHR$(CONVER)
15140 RETURN

************************************************************************
*                                                                      *
*   NAME:           END WEATHER                                        *
*   FUNCTION:       END PROGRAM WHEN END OF WEATHER FILE IS REACHED    *
*   CALLED FROM:    MAIN -- MAIN PROGRAM                               *
*   CALLS:          NO OTHER ROUTINES                                  *
*                                                                      *
************************************************************************

16000 CLOSE: CLS : PRINT"END OF WEATHER FILE.....
16010 END
```

```
1  REM    *****************************************************************
2  REM    *                                                               *
3  REM    *        NAME:          weather                                 *
4  REM    *        FUNCTION:      produces weather file for refined model *
5  REM    *        CALLED FROM:   this is main program                    *
6  REM    *        CALLS:         stabclass, check, output                *
7  REM    *                                                               *
8  REM    *****************************************************************
9  REM
10 DIM ENP$(80), FE$(8)
20 DIM TC(12), CS(12), DC(12), SA(12), mo$(12)
25 PASS = 1: pi = 3.14159: oh = -1
26 WD$ = "27": REM if first hour is calm assumed wind direction will be W.
30 FOR Z = 1 TO 12: READ mo$(Z): NEXT Z: REM names of months
35 DATA "JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC"
40 FOR Z = 1 TO 12: READ TC(Z): NEXT Z: REM daily change in equation of time
50 DATA -20.4,1.8888,16.26666,14.17241,-.6,-11.931034,-5.53,11.5,20.62,12.43,
   -9.965517,-27.8
60 FOR Z = 1 TO 12: READ CS(Z):  NEXT Z: REM equation of time at start of month
70 DATA -192, -813,-754,-248,171,145,-213,-376,-12,605,980,671
80 FOR Z = 1 TO 12: READ DC(Z):  NEXT Z: REM daily change in solar declination
90 DATA .189,.3275,.368,.3506896,.232,.0534482,-.163333,-.3193333,-.3668965,
   -.366,-.247931,-.055
100 FOR Z = 1 TO 12: READ SA(Z): NEXT Z: REM solar declination at start of month
110 DATA -23.04,-17.2,-7.49,4.18,14.54,21.58,23.09,18.0,8.3,-2.57,-14.14,-21.43
115 FI$ = "WX000A.DAT"
116 INPUT "LATITUDE OF STATION IN DEGREES (e.g. 39.54) "; la
117 INPUT "LONGITUDE OF STATION IN DEGREES (e.g. 112.4)"; LO
120 OPEN "C:" + FI$ FOR INPUT AS #1
121 IF PASS > 1 THEN 127
125 OPEN "C:WEATHER1.DAT" FOR OUTPUT AS #2
126 GOTO 130
127 OPEN "C:WEATHER1.DAT" FOR APPEND AS #2
130 REM: IF EOF(1) THEN GOTO 300
140 INPUT #1, ENP$: REM read NCS weather data
150 FE$ = MID$(ENP$, 1, 8)
160 IF FE$ = "99998888" THEN GOTO 300: REM end of file
165 da$ = MID$(ENP$, 10, 2)
170 ws$ = MID$(ENP$, 41, 2): ws = VAL(ws$)
175 yr$ = "19" + MID$(ENP$, 6, 2)
180 wd1$ = MID$(ENP$, 39, 2): wd1 = VAL(wd1$): IF wd1 > 0 THEN WD$ = wd1$
181 REM retain last direction when wind calm (36 is north, 00 is calm)
185 WD = VAL(WD$): WDR = INT((WD - .35 + RND(1)) * 10):rem randomize last digit
186 IF WDR < 0 THEN WDR = WDR + 360
187 IF WDR > 360 THEN WDR = WDR - 360
188 WDR$ = RIGHT$("   " + STR$(WDR), 3)
190 db$ = MID$(ENP$, 47, 3): db = VAL(db$)
200 hr$ = MID$(ENP$, 12, 2): hr = VAL(hr$)
205 CE$ = MID$(ENP$, 14, 3): CE = 100 * VAL(CE$)
207 IF CE$ <> "---" THEN CE$ = STR$(CE)
201 IF CE$ = "---" THEN CE = 99800: CE$ = "   "
210 SC = 0
211 FOR II = 1 TO 4
212 SC$ = MID$(ENP$, 16 + II, 1): SC1 = VAL(SC$)
```

```
213 IF SC$ = "X" THEN SC1 = 10
214 IF SC1 > SC THEN SC = SC1
215 NEXT II

216 IF SC < 4 THEN SC = 4: GOTO 220
217 IF SC > 6 THEN SC = 10: GOTO 220
218 IF SC > 3 AND SC < 7 THEN SC = 7
220 mo = VAL(MID$(ENP$, 8, 2))
230 da = VAL(MID$(ENP$, 10, 2))
240 GOSUB 400: REM stabclass -- compute atmospheric stability class
245 GOSUB 1000: REM check -- checks for zero values
250 GOSUB 800: REM output -- print out weather record for dispersion
255 x = FRE(x$): REM housecleaning
260 GOTO 130
300 CLOSE #1
310 CLOSE #2
315 PASS = PASS + 1
320 IF PASS = 2 THEN FI$ = "WX000B.DAT"
325 IF PASS = 3 THEN FI$ = "WX000C.DAT"
330 IF PASS = 4 THEN FI$ = "WX000D.DAT"
340 IF PASS = 5 THEN END
350 GOTO 120
351  REM
```

```
360  REM  ********************************************************************
361  REM  *                                                                  *
362  REM  *        NAME:         stabclass                                   *
363  REM  *        FUNCTION:     assigns P/G stability class to hour         *
364  REM  *        CALLED FROM: main                                         *
365  REM  *        CALLS:        overcast, clouds                            *
366  REM  *                                                                  *
367  REM  ********************************************************************
368  REM
400 DI = da * TC(mo): REM interpolation of eouation of time for day
410 CT = CS(mo) + DI: REM add in equation of time value at start of month
420 LC = (15 * INT(LO / 15) - LO) * 24 / 360: REM longitude correction for time
430 TA = hr - CT / 3600: REM TA is local apparent time in hours
440 ti = (12 - TA) / 24 * 360: REM convert time to hour angle
450 ti = ti * pi / 180: REM convert hour angle to radians
460 DI = da * DC(mo): DCL = SA(mo) + DI: REM interpolate solar declination
470 lr = la * pi / 180: REM convert latitude to radians
480 dr = DCL * pi / 180: REM convert deciination to radians
490 AD = (SIN(dr) * SIN(lr)) + (COS(dr) * COS(lr) * COS(ti))
495 REM AD = sin(sun angle)
500 h = ATN(AD / SQR(1 - AD * AD)): REM arcsine function.  H=sun angle
510 h = h * 180 / pi: REM convert sun angle to degrees.
525 REM: insolation class as a function of sun angle.
520 IF h <= 0 THEN ins = 0: GOTO 580: REM night
530 IF h > 0 AND h <= 15 THEN ins = 1
540 IF h > 15 AND h <= 35 THEN ins = 2
550 IF h > 35 AND h <= 60 THEN ins = 3
560 IF h > 60 THEN ins = 4
561 IF SC = 10 THEN GOSUB 1200:rem overcast
562 IF SC >= 5 AND SC < 10 THEN GOSUB 1250:rem clouds
563 IF ins = -1 THEN PG = 4: ins = 1: RETURN: REM LOW OVERCAST
565 REM: Pasquill-Gifford class based on weather, insolation.
580 IF ins = 4 AND ws < 6 THEN PG = 1: RETURN
590 IF ins = 4 AND ws < 10 AND ws >= 6 THEN PG = 2: RETURN
600 IF ins = 4 AND ws >= 10 THEN PG = 3: RETURN
610 IF ins = 3 AND ws < 2 THEN PG = 1: RETURN
```

```
620 IF ins = 3 AND ws < 8 AND ws >= 2 THEN PG = 2: RETURN
630 IF ins = 3 AND ws >= 8 AND ws < 12 THEN PG = 3: RETURN
640 IF ins = 3 AND ws >= 12 THEN PG = 4: RETURN
650 IF ins = 2 AND ws < 4 THEN PG = 2: RETURN
660 IF ins = 2 AND ws >= 4 AND ws < 10 THEN PG = 3: RETURN
670 IF ins = 2 AND ws >= 10 THEN PG = 4: RETURN
680 IF ins = 1 AND ws < 4 THEN PG = 3: RETURN
690 IF ins = 1 AND ws >= 4 THEN PG = 4: RETURN
700 IF ins = 0 AND ws <= 6 AND SC >= 5 THEN PG = 5: RETURN
710 IF ins = 0 AND ws > 6 AND SC >= 5 THEN PG = 4: RETURN
720 IF ins = 0 AND ws <= 10 AND SC < 5 THEN PG = 5: RETURN
730 IF ins = 0 AND ws > 10 AND SC < 5 THEN PG = 4: RETURN
740 RETURN
780  REM
781  REM  *******************************************************************
782  REM  *                                                                 *
783  REM  *          NAME:         output                                   *
784  REM  *          FUNCTION:     writes weather file                      *
785  REM  *          CALLED FROM: main                                      *
786  REM  *          CALLS:        missing                                  *
787  REM  *                                                                 *
788  REM  *******************************************************************
789  REM
800 PRINT #2, yr$; mo$(mo); da$; hr$; db$; ws$; WDR$; PG$
805 GOSUB 1300:rem missing
810 RETURN
981 REM
982 REM  *******************************************************************
983 REM  *                                                                 *
984 REM  *          NAME:         check                                    *
985 REM  *          FUNCTION:     checks for zero values                   *
986 REM  *          CALLED FROM: main                                      *
987 REM  *          CALLS:        zero                                     *
988 REM  *                                                                 *
989 REM  *******************************************************************
990 REM
1000 W$ = yr$: GOSUB 1100: yr$ = W$: REM subroutine zero replaces blanks
1010 W$ = mo$(mo): GOSUB 1100: mo$(mo) = W$
1020 W$ = da$: GOSUB 1100: da$ = W$
1025 W$ = hr$: GOSUB 1100: hr$ = W$
1030 W$ = db$: GOSUB 1100: db$ = W$
1040 W$ = ws$: GOSUB 1100: ws$ = W$
1050 W$ = WDR$: GOSUB 1100: WDR$ = W$
1060 PG$ = RIGHT$(STR$(PG), 1)
1070 W$ = PG$: GOSUB 1100: PG$ = W$
1080 RETURN
1081 REM
```

```
1082 REM  ******************************************************************
1083 REM  *                                                                *
1084 REM  *        NAME:         zero                                      *
1085 REM  *        FUNCTION:     replaces blanks with zeros                *
1086 REM  *        CALLED FROM:  main                                      *
1087 REM  *        CALLS:        no other routines                         *
1088 REM  *                                                                *
1089 REM  ******************************************************************

1090 REM
1100 FOR I = 1 TO LEN(W$): IF MID$(W$, I, 1) = " " THEN MID$(W$, I, 1) = "0"
1110 NEXT I
1120 RETURN
1280 REM
1182 REM  ******************************************************************
1183 REM  *                                                                *
1184 REM  *        NAME:         overcast                                  *
1185 REM  *        FUNCTION:     adjusts insolation for overcast           *
1186 REM  *        CALLED FROM:  stabclass                                 *
1187 REM  *        CALLS:        no other routines                         *
1188 REM  *                                                                *
1189 REM  ******************************************************************
1190 REM
1200 IF CE < 7000 THEN ins = -1: RETURN: rem flag for low overcast
1210 IF CE >= 7000 AND CE <= 16000 THEN ins = ins - 2
1220 IF CE > 16000 THEN ins = ins - 1
1225 IF ins < 1 THEN ins = 1
1226 RETURN
1228 REM
1230 REM  ******************************************************************
1233 REM  *                                                                *
1234 REM  *        NAME:         clouds                                    *
1235 REM  *        FUNCTION:     adjusts insolation for clouds             *
1236 REM  *        CALLED FROM:  stabclass                                 *
1237 REM  *        CALLS:        no other routines                         *
1238 REM  *                                                                *
1239 REM  ******************************************************************
1240 REM
1250 IF CE < 7000 THEN ins = ins - 2
1260 IF CE >= 7000 AND CE <= 16000 THEN ins = ins - 1
1265 IF ins < 1 THEN ins = 1
1270 RETURN
```

```
1330 REM  ******************************************************************
1333 REM  *                                                                *
1334 REM  *        NAME:         missing                                   *
1335 REM  *        FUNCTION:     finds hours of missing data               *
1336 REM  *        CALLED FROM:  output                                    *
1337 REM  *        CALLS:        no other routines                         *
1338 REM  *                                                                *
1339 REM  ******************************************************************
1340 REM
1300 oh = oh + 1: IF oh = 24 THEN oh = 0
1310 IF hr > oh THEN LPRINT "missing data ", od$, om$, oh: GOTO 1300
1315 od$ = da$: oh$ = hr$: om$ = mo$(mo)
1320 RETURN
```

```
4  REM   ******************************************************************
5  REM   *                                                                *
6  REM   *            NAME:         preproc                               *
7  REM   *            FUNCTION:     subsets weather file over desired interval  *
8  REM   *            CALLED FROM: this is main program                    *
9  REM   *            CALLS:        no other routines                     *
10 REM   *                                                                *
11 REM   ******************************************************************
12 REM
13 DIM MO(12), HR(24), DA(31)
18 PRINT
20 INPUT "Enter starting time: month (1-12), day (1-31), hour (0-23)"; MONTH,
   DAY, HOUR
25 PRINT
30 IF MONTH = 13 THEN END
40 YEAR = 82
50 GOSUB 190:rem julien
60 START = (24 * MO(MONTH - 1)) + ((DAY - 1) * 24) + HOUR + 1
70 INPUT "Enter ending time: month (1-12), day (1-31), hour (0-23)"; MONTH, DAY,
   HOUR
80 EN = (24 * MO(MONTH - 1)) + ((DAY - 1) * 24) + HOUR + 1
100 OPEN "WEATHER1.DAT" FOR RANDOM AS #1 LEN = 22
105 OPEN "WEATHER.DAT" FOR OUTPUT AS #2
110 FIELD 1, 21 AS NP$
120 FOR A = START TO EN
130 GET #1, A
145 PRINT #2, NP$
160 NEXT A
170 CLOSE
180 END
181 REM
```

```
182 REM     *****************************************************************
183 REM     *                                                               *
184 REM     *          NAME:         julien                                 *
185 REM     *          FUNCTION:     relates julien day and month           *
186 REM     *          CALLED FROM:  main                                   *
187 REM     *          CALLS:        no other routines                      *
188 REM     *                                                               *
189 REM     *****************************************************************
190 REM
191 MO(1) = 31
200 MO(2) = 59
210 MO(3) = 90
220 MO(4) = 120
230 MO(5) = 151
240 MO(6) = 181
250 MO(7) = 212
260 MO(8) = 243
270 MO(9) = 273
280 MO(10) = 304
290 MO(11) = 334
300 MO(12) = 365
305 rem:leap year
310 IF YEAR / 4 = INT(YEAR/4) AND MONTH >2 THEN MO(MONTH-1)=MO(MONTH-1)
320 RETURN
```